Nordic Journal of Computing

A Bisimulation Method for Cryptographic Protocols

Martín Abadi
Systems Research Center
Compaq
ma@pa.dec.com

Andrew D. Gordon
Microsoft Research
adg@microsoft.com

Abstract. We introduce a definition of bisimulation for cryptographic protocols. The definition includes a simple and precise model of the knowledge of the environment with which a protocol interacts. Bisimulation is the basis of an effective proof technique, which yields proofs of classical security properties of protocols and also justifies certain protocol optimizations. The setting for our work is the spi calculus, an extension of the pi calculus with cryptographic primitives. We prove the soundness of the bisimulation proof technique within the spi calculus.

1. Introduction

In reasoning about a reactive system, it is necessary to consider not only the steps taken by the system but also the steps taken by its environment. In the case where the reactive system is a cryptographic protocol, the environment may well be hostile, so little can be assumed about its behaviour. Therefore, the environment may be modelled as a nondeterministic process capable of intercepting messages and of sending any message that it can construct at any point. This approach to describing the environment is fraught with difficulties; the resulting model can be somewhat arbitrary, hard to understand, and hard to reason about.

Bisimulation techniques [15, 17] provide an alternative approach. Basically, using bisimulation techniques, we can equate two systems whenever we can establish a correspondence between their steps. We do not need to describe the environment explicitly, or to analyze its possible internal computations.

Bisimulation techniques have been applied in a variety of areas and under many guises. Their application to cryptographic protocols, however, presents new challenges.

- Consider, for example, a secure communication protocol $P(M)$, where some cleartext M is transmitted encrypted under a session key. We may like to argue that $P(M)$ preserves the secrecy of M, and may want to express this secrecy property by saying that $P(M)$ and $P(N)$ are equivalent, for every M and N. This equivalence may be sensible because, although $P(M)$ and $P(N)$ send different messages, an attacker that does not have the session key cannot identify the cleartext. Unfortunately, a standard notion of bisimulation would require that $P(M)$

Received July 1998. Revised October 1998.

and $P(N)$ send identical messages. So we should relax the definition of bisimulation to permit indistinguishable messages.

- In reasoning about a protocol, we need to consider its behaviour in reaction to inputs from the environment. These inputs are not entirely arbitrary. For example, consider a system $P(M)$ which discloses M when it receives a certain password. Assuming that the password remains secret, $P(M)$ should be equivalent to $P(N)$. In order to argue this, we need to characterize the set of possible inputs from the environment, and to show that it cannot include the password.
- Two messages that are indistinguishable at one point in time may become distinguishable later on. In particular, the keys under which they are encrypted may be disclosed to the environment, which may then inspect the cleartext that these keys were concealing. Thus, the notion of indistinguishability should be sensitive to future events.

  Conversely, the set of possible inputs from the environment grows with time, as the environment intercepts messages and learns values that were previously secret.

In short, a definition of bisimulation for cryptographic protocols should explain what outputs are indistinguishable for the environment, and what inputs the environment can generate at any point in time. In this paper we introduce a definition of bisimulation that provides the necessary account of the knowledge of the environment. As we show, bisimulation can be used for reasoning about examples like those sketched informally above. More generally, bisimulation can be used for proving authenticity and secrecy properties of protocols, and also for justifying certain protocol optimizations.

We develop our bisimulation proof technique in the context of the spi calculus [1, 2, 3, 4], an extension of the pi calculus [16] with cryptographic primitives. For simplicity, we consider only shared-key cryptography, although we believe that public-key cryptography could be treated through similar methods. Within the spi calculus, we prove the soundness of our technique. More precisely, we prove that bisimulation yields a sufficient condition for testing equivalence, the relation that we commonly use for specifications in the spi calculus.

We have developed other proof techniques for the spi calculus in earlier work. The one presented in this paper is a useful addition to our set of tools. Its distinguishing characteristic is its kinship to bisimulation proof techniques for other classes of systems. In particular, bisimulation proofs can often be done without creativity, essentially by state-space exploration.

The next section is a review of the spi calculus. Section 3 describes our proof method and Section 4 illustrates its use through some small examples. Section 5 discusses related work. Section 6 concludes. The Appendix contains proofs.

2. The Spi Calculus (Review)

This section reviews the spi calculus, borrowing from earlier presentations. It gives the syntax and informal semantics of the spi calculus, introduces the main notations for its operational semantics, and finally defines testing equivalence.

2.1 Syntax

We assume an infinite set of *names* and an infinite set of *variables*. We let c, m, n, p, q, and r range over names, and let w, x, y, and z range over variables. When they represent keys, we let k and K range over names too.

The set of *terms* is defined by the grammar:

$L, M, N ::=$	terms
n	name
(M, N)	pair
0	zero
$suc(M)$	successor
$\{M\}_N$	shared-key encryption
x	variable

Intuitively, $\{M\}_N$ represents the ciphertext obtained by encrypting the term M under the key N using a shared-key cryptosystem such as DES [6]. In examples, we write 1 as a shorthand for the term $suc(0)$.

The set of *processes* is defined by the grammar:

$P, Q, R ::=$	processes
$\overline{M}\langle N\rangle.P$	output
$M(x).P$	input
$P \mid Q$	composition
$(\nu n)P$	restriction)
$!P$	replication
$[M \text{ } is \text{ } N] \, P$	match
$\mathbf{0}$	nil
$let \ (x, y) = M \ in \ P$	pair splitting
$case \ M \ of \ 0 : P \ suc(x) : Q$	integer case
$case \ L \ of \ \{x\}_N \ in \ P$	decryption

The scopes of bound variables and names are as follows. In $M(x).P$, the scope of x is P. In $(\nu n)P$, the scope of n is P. In $let \ (x, y) = M \ in \ P$, the scope of x and y is P. In $case \ M \ of \ 0 : P \ suc(x) : Q$, the scope of x is Q. In $case \ L \ of \ \{x\}_N \ in \ P$, the scope of x is P.

We abbreviate $\overline{M}\langle N\rangle.\mathbf{0}$ to $\overline{M}\langle N\rangle$. We write $P[M/x]$ for the outcome of replacing each free occurrence of x in process P with the term M, and identify processes up to renaming of bound variables and names. Intuitively, processes have the following meanings:

- An *output process* $\overline{M}\langle N\rangle.P$ is ready to output N on M, and then to behave as P. The output happens only when there is a process ready to input from M. An *input process* $M(x).Q$ is ready to input from M, and then to behave as $Q[N/x]$, where N is the input received.
- A *composition* $P \mid Q$ behaves as P and Q running in parallel.
- A *restriction* $(\nu n)P$ is a process that makes a new, private name n, which may occur in P, and then behaves as P.
- A *replication* $!P$ behaves as infinitely many replicas of P running in parallel.
- A *match* $[M\, is\, N]\,P$ behaves as P provided that M and N are the same term; otherwise it is stuck, that is, it does nothing.
- The *nil* process $\mathbf{0}$ does nothing.
- A *pair splitting* process $let\ (x, y) = M\ in\ P$ behaves as $P[N/x][L/y]$ if M is a pair (N, L), and it is stuck if M is not a pair.
- An *integer case* process $case\ M\ of\ 0 : P\ suc(x) : Q$ behaves as P if M is 0, as $Q[N/x]$ if M is $suc(N)$ for some N, and otherwise is stuck.
- A *decryption* process $case\ L\ of\ \{x\}_N\ in\ P$ attempts to decrypt L with the key N. If L has the form $\{M\}_N$, then the process behaves as $P[M/x]$. Otherwise the process is stuck.

For example, $P \triangleq m(x).case\ x\ of\ \{y\}_K\ in\ \overline{m}\langle\{0\}_y\rangle$ is a process that is ready to receive a message x on the channel m. When the message is a ciphertext of the form $\{y\}_K$, process P sends 0 encrypted under y on the channel m. This process may be put in parallel with a process $Q \triangleq \overline{m}\langle\{K'\}_K\rangle$, which sends the name K' encrypted under K on the channel m. In order to restrict the use of K to P and Q, we may form $(\nu K)(P \mid Q)$. The environment of $(\nu K)(P \mid Q)$ will not be able to construct any message of the form $\{y\}_K$, since K is bound. Therefore, the component P of $(\nu K)(P \mid Q)$ may output $\{0\}_{K'}$, but not $\{0\}_z$ for any z different from K'. Alternatively, the component P of $(\nu K)(P \mid Q)$ may produce no output: for example, were it to receive 0 on the channel m, it would get stuck.

We write $fn(M)$ and $fn(P)$ for the sets of names free in term M and process P respectively, and write $fv(M)$ and $fv(P)$ for the sets of variables free in M and P respectively. A term or process is *closed* if it has no free variables. *Proc* is the set of closed processes.

2.2 Operational Semantics

An *abstraction* is an expression of the form $(x)P$, where x is a bound variable and P is a process. Intuitively, $(x)P$ is like the process $p(x).P$ minus the name p. A *concretion* is an expression of the form $(\nu m_1, \ldots, m_k)\langle M\rangle P$, where M is a term, P is a process, $k \geq 0$, and the names $m_1, \ldots, m_k$ are bound in M and P. Intuitively, $(\nu m_1, \ldots, m_k)\langle M\rangle P$ is like the process $(\nu m_1)\ldots(\nu m_k)\overline{p}\langle M\rangle P$ minus the name p, provided p is not one of $m_1, \ldots,$

m_k. We often write concretions as $(\nu\vec{m})\langle M\rangle P$, where $\vec{m} = m_1, \ldots, m_k$, or simply $(\nu)\langle M\rangle P$ if $k = 0$. Finally, an *agent* is an abstraction, a process, or a concretion. We use the metavariables A and B to stand for arbitrary agents, and let $fv(A)$ and $fn(A)$ be the sets of free variables and free names of an agent A, respectively.

A *barb* is a name m (representing input) or a co-name $\overline{m}$ (representing output). An *action* is a barb or the distinguished *silent action* τ. The *commitment relation* is written $P \xrightarrow{\alpha} A$, where P is a closed process, α is an action, and A is a closed agent. The exact definition of commitment appears in earlier papers on the spi calculus [3, 4], and is reviewed in the Appendix; informally, the definition says:

- $P \xrightarrow{\tau} Q$ means that P becomes Q in one silent step (a τ step).
- $P \xrightarrow{m} (x)Q$ means that, in one step, P is ready to receive an input x on m and then to become Q.
- $P \xrightarrow{\overline{m}} (\nu m_1, \ldots, m_k)\langle M\rangle Q$ means that, in one step, P is ready to create the new names m_1, …, m_k, to send M on m, and then to become Q.

2.3 Testing Equivalence

We say that two closed processes P and Q are testing equivalent, and write $P \simeq Q$, when for every closed process R and every barb β, if

$$P \mid R \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A, then

$$Q \mid R \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B, and vice versa.

For example, the processes $(\nu K)\overline{m}\langle\{0\}_K\rangle$ and $(\nu K)\overline{m}\langle\{1\}_K\rangle$ are testing equivalent. We may interpret this equivalence as a security property, namely that the process $(\nu K)\overline{m}\langle\{x\}_K\rangle$ does not reveal to its environment whether x is 0 or 1. In the examples contained in Section 4 and in earlier papers, various other properties (including, in particular, secrecy properties) are formulated in terms of testing equivalence.

In this paper we develop a sound technique for proving testing equivalence: we introduce a definition of bisimulation and show that if two closed processes are in one of our bisimulation relations then they are testing equivalent. In fact, as we discuss, they are also equivalent in a stronger sense.

3. Framed Bisimulation

This section defines our notion of bisimulation, which we call *framed bisimulation*.

3.1 Frames and Theories

Our definition of bisimulation is based on the notions of a *frame* and of a *theory*. A bisimulation does not simply relate two processes P and Q, but instead relates two processes P and Q in the context of a frame and a theory. The frame and the theory represent the knowledge of the environment of P and Q.

- A frame is a finite set of names. Intuitively, a frame is a set of names available to the environment of the processes P and Q. We use fr to range over frames.
- A theory is a finite set of pairs of terms. Intuitively, a theory that includes a pair (M, N) indicates that the environment cannot distinguish the data M coming from process P and the data N coming from process Q. We use th to range over theories.

Next we define the predicate $(fr, th) \vdash M \leftrightarrow N$ inductively, by a set of rules. Intuitively, this predicate means that the environment cannot distinguish M coming from P and N coming from Q, and that the environment has (or can construct) M in interaction with P and N in interaction with Q.

(Eq Frame)
$$\frac{n \in fr}{(fr, th) \vdash n \leftrightarrow n}$$

(Eq Theory)
$$\frac{(M, N) \in th}{(fr, th) \vdash M \leftrightarrow N}$$

(Eq Variable)
$$\frac{}{(fr, th) \vdash x \leftrightarrow x}$$

(Eq Pair)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N'}{(fr, th) \vdash (M, N) \leftrightarrow (M', N')}$$

(Eq Zero)
$$\frac{}{(fr, th) \vdash 0 \leftrightarrow 0}$$

(Eq Suc)
$$\frac{(fr, th) \vdash M \leftrightarrow M'}{(fr, th) \vdash suc(M) \leftrightarrow suc(M')}$$

(Eq Encrypt)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N'}{(fr, th) \vdash \{M\}_N \leftrightarrow \{M'\}_{N'}}$$

For example, if $fr = \{n\}$ and $th = \{(\{0\}_K, \{n\}_K)\}$, where n and K are distinct names, then we have $(fr, th) \vdash n \leftrightarrow n$ and $(fr, th) \vdash \{0\}_K \leftrightarrow \{n\}_K$, and also $(fr, th) \vdash (n, \{0\}_K) \leftrightarrow (n, \{n\}_K)$, but we have neither $(fr, th) \vdash K \leftrightarrow K$ nor $(fr, th) \vdash \{n\}_K \leftrightarrow \{0\}_K$.

We say that the pair (fr, th) is ok, and write $(fr, th) \vdash ok$, if two conditions hold:

(1) whenever $(M, N) \in th$:

- M is closed and there are terms M_1 and M_2 such that $M = \{M_1\}_{M_2}$ and there is no N_2 such that $(fr, th) \vdash M_2 \leftrightarrow N_2$;

   ◦ N is closed and there are terms N_1 and N_2 such that $N = \{N_1\}_{N_2}$ and there is no M_2 such that $(fr, th) \vdash M_2 \leftrightarrow N_2$;

(2) whenever $(M, N) \in th$ and $(M', N') \in th$, $M = M'$ if and only if $N = N'$.

Intuitively, the first condition requires that each term in a pair (M, N) in a theory be formed by ciphertexts that the environment cannot decrypt. For example, the requirement that there be no N_2 such that $(fr, th) \vdash M_2 \leftrightarrow N_2$ means that the environment cannot construct M_2 for decrypting M. The second condition guarantees that no ciphertext is equated to two other ciphertexts. This condition is essential because the environment can compare ciphertexts even when it cannot decrypt them (see Example 2 of Section 4).

3.2 Ordering Frame-Theory Pairs

We define an ordering between pairs of frames and theories as follows: we let $(fr, th) \leq (fr', th')$ if and only if for all M and N, $(fr, th) \vdash M \leftrightarrow N$ implies $(fr', th') \vdash M \leftrightarrow N$. This relation is reflexive and transitive. It is not the same as the pairwise ordering induced by subset inclusion; $fr \subseteq fr'$ and $th \subseteq th'$ imply $(fr, th) \leq (fr', th')$, but the converse implication does not hold.

PROPOSITION 1. *Suppose $(fr', th') \vdash ok$. Then $(fr, th) \leq (fr', th')$ if and only if $fr \subseteq fr'$ and $(fr', th') \vdash M \leftrightarrow N$ for each $(M, N) \in th$.*

PROOF. To show this right-to-left, suppose that $fr \subseteq fr'$ and $(fr', th') \vdash M \leftrightarrow N$ for each $(M, N) \in th$. Consider any M and N such that $(fr, th) \vdash M \leftrightarrow N$. An induction on the derivation of $(fr, th) \vdash M \leftrightarrow N$ shows that $(fr', th') \vdash M \leftrightarrow N$. Therefore $(fr, th) \leq (fr', th')$.

For the left-to-right direction, suppose that $(fr, th) \leq (fr', th')$. By (Eq Theory), $(fr, th) \vdash M \leftrightarrow N$ for any $(M, N) \in th$. So $(fr, th) \leq (fr', th')$ implies $(fr', th') \vdash M \leftrightarrow N$. Consider any $n \in fr$. We have $(fr, th) \vdash n \leftrightarrow n$ by (Eq Frame), and then $(fr, th) \leq (fr', th')$ implies $(fr', th') \vdash n \leftrightarrow n$. The latter judgment can only be derived from (Eq Frame), since $(fr', th') \vdash ok$, so $n \in fr'$. Since n was arbitrary, $fr \subseteq fr'$. □

As indicated above, we view a pair (fr, th) as a representation for the knowledge of an environment. With this view, and assuming that $(fr', th') \vdash ok$, the relation $(fr, th) \leq (fr', th')$ means that the environment may go from the knowledge represented in (fr, th) to the knowledge represented in (fr', th'). The definition of $(fr, th) \leq (fr', th')$ implies that the set of names and terms that the environment has (or can construct) grows in this transition. It also implies that any indistinguishable pair of terms remains indistinguishable after the transition. So, if ever we assert that (fr, th) characterizes an environment, we should take care that (fr, th) does not imply that the environment cannot distinguish two terms M and N if later information would allow the environment to distinguish these terms. For example, if fr' includes the name n, then th should not contain $(\{0\}_n, \{1\}_n)$. Intuitively, the acquisition of the name n would allow the environment to distinguish $\{0\}_n$ and $\{1\}_n$, so $(fr', th') \vdash \{0\}_n \leftrightarrow \{1\}_n$ would not hold. On the other hand, th may contain $(\{0\}_n, \{0\}_n)$; in that case, th' could not contain $(\{0\}_n, \{0\}_n)$, but we would at least have $(fr', th') \vdash \{0\}_n \leftrightarrow \{0\}_n$.

3.3 Framed Relations and Bisimulations

For a theory th, we let $fn(th) = \bigcup\{fn(M) \cup fn(N) \mid (M,N) \in th\}$. We let $\pi_1(th) = \{M \mid (M,N) \in th\}$ and $\pi_2(th) = \{N \mid (M,N) \in th\}$, and write $fn(\pi_1(th))$ and $fn(\pi_2(th))$ for the sets of names $\bigcup\{fn(M) \mid M \in \pi_1(th)\}$ and $\bigcup\{fn(N) \mid N \in \pi_2(th)\}$ respectively.

A *framed process pair* is a quadruple (fr, th, P, Q) such that P and Q are closed processes, fr is a frame, and th is a theory. When $\mathcal{R}$ is a set of framed process pairs, we write $(fr, th) \vdash P \; \mathcal{R} \; Q$ to mean $(fr, th, P, Q) \in \mathcal{R}$. A *framed relation* is a set $\mathcal{R}$ of framed process pairs such that $(fr, th) \vdash ok$ whenever $(fr, th) \vdash P \; \mathcal{R} \; Q$.

A *framed simulation* is a framed relation $\mathcal{S}$ such that, whenever $(fr, th) \vdash P \; \mathcal{S} \; Q$, the following three conditions hold.

- If $P \xrightarrow{\tau} P'$
  then there is a process Q' such that $Q \xrightarrow{\tau} Q'$ and $(fr, th) \vdash P' \; \mathcal{S} \; Q'$.
- If $P \xrightarrow{c} (x)P'$ and $c \in fr$
  then there is an abstraction $(x)Q'$ such that $Q \xrightarrow{c} (x)Q'$ and, for all sets $\{\vec{n}\}$ disjoint from $fn(P) \cup fn(Q) \cup fr \cup fn(th)$ and all closed M and N, if $(fr \cup \{\vec{n}\}, th) \vdash M \leftrightarrow N$ then $(fr \cup \{\vec{n}\}, th) \vdash P'[M/x] \; \mathcal{S} \; Q'[N/x]$.
- If $P \xrightarrow{\overline{c}} (\nu\vec{m})\langle M\rangle P'$, $c \in fr$, and the set $\{\vec{m}\}$ is disjoint from $fn(P) \cup fn(\pi_1(th)) \cup fr$
  then there is a concretion $(\nu\vec{n})\langle N\rangle Q'$ such that $Q \xrightarrow{\overline{c}} (\nu\vec{n})\langle N\rangle Q'$ and the set $\{\vec{n}\}$ is disjoint from $fn(Q) \cup fn(\pi_2(th)) \cup fr$, and there is a frame-theory pair (fr', th') such that $(fr, th) \leq (fr', th')$, $(fr', th') \vdash M \leftrightarrow N$, and $(fr', th') \vdash P' \; \mathcal{S} \; Q'$.

We may explain these conditions as follows.

- The first condition simply requires that if P can take a τ step then Q can match this step.
- The second condition concerns input steps where the channel c on which the input happens is in fr (that is, it is known to the environment). In this case, we must consider the possible inputs M from the environment to $(x)P'$, namely the terms M that the environment can construct according to $(fr \cup \{\vec{n}\}, th)$. The names in $\vec{n}$ are fresh names, intuitively names just generated by the environment. Correspondingly, we consider the possible inputs N for $(x)Q'$, for an appropriate $(x)Q'$ obtained from Q. We then require that giving these inputs to $(x)P'$ and $(x)Q'$, respectively, yields related processes $P'[M/x]$ and $Q'[N/x]$.

The choice of $(x)Q'$ is independent of the choices of M and N. So, in the technical jargon, we may say that $\mathcal{S}$ is a *late* framed simulation.

- The third condition concerns output steps where the channel c on which the output happens is in *fr* (that is, it is known to the environment). In this case, P outputs the term M while creating the names $\vec{m}$. The condition requires that Q can output a corresponding term N while creating some names $\vec{n}$. It also constrains M and N, and the resulting processes P' and Q'. The constraints concern a new frame-theory pair $(\mathit{fr}', \mathit{th}')$. Intuitively, this pair represents the knowledge of the environment after the output step. The requirement that $(\mathit{fr}', \mathit{th}') \vdash M \leftrightarrow N$ means that the environment obtains M in interaction with P and N in interaction with Q, and that it should not be able to distinguish them from one another.

  Because we do not impose a minimality requirement on $(\mathit{fr}', \mathit{th}')$, this pair may attribute "too much" knowledge to the environment. For example, fr' may contain names that are neither in *fr* nor in M or N, so intuitively the environment would not be expected to know these names. On the other hand, the omission of a minimality requirement results in simpler definitions, and does not compromise soundness.

A *framed bisimulation* is a framed relation $\mathcal{S}$ such that both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are framed simulations. *Framed bisimilarity* (written $\sim_f$) is the greatest framed bisimulation. By the Knaster-Tarski fixpoint theorem, since the set of framed relations ordered by subset inclusion forms a complete lattice, framed bisimilarity exists, and equals the union of all framed bisimulations.

Our intent is that our definition of framed bisimulation may serve as the basis for an algorithm, at least for finite-state processes. Unfortunately, the definition contains several levels of quantification. The universal quantifiers present a serious obstacle to any algorithm for constructing framed bisimulations. In particular, the condition for input steps concerns all possible inputs M and N; these inputs are of unbounded size, and may contain an arbitrary number of fresh names. However, we conjecture that the inputs can be classified according to a finite number of patterns—intuitively, because the behaviour of any finite-state process can depend on at most a finite portion of its inputs. An algorithm for constructing framed bisimulations might consider all inputs that match the same pattern at once. We leave the invention of such an algorithm for future work.

3.4 Soundness

Our main soundness theorem about framed bisimulation is that it is a sufficient condition for testing equivalence.

THEOREM 1. *Consider any closed processes P and Q, and any name $n \notin \mathit{fn}(P) \cup \mathit{fn}(Q)$. Suppose that $(\mathit{fn}(P) \cup \mathit{fn}(Q) \cup \{n\}, \emptyset) \vdash P \sim_f Q$. Then $P \simeq Q$.*

This theorem implies that if we want to prove that two processes are testing equivalent, then we may construct a framed bisimulation $\mathcal{S}$ such that $(fn(P) \cup fn(Q) \cup \{n\}, \emptyset) \vdash P \;\mathcal{S}\; Q$ where n is a single, arbitrary new name. (The addition of the name n is technically convenient, but may not be necessary.) The next section illustrates this approach through several examples.

The proof of this theorem requires a number of auxiliary notations, definitions, and lemmas. The details of the proof are in the Appendix; here we only indicate its main idea. First we extend the relation $\leftrightarrow$ to processes: we define the predicate $(fr, th) \vdash P \leftrightarrow Q$ by the following rules.

(Eq Out)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N' \quad (fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash \overline{M}\langle N\rangle.P \leftrightarrow \overline{M'}\langle N'\rangle.P'}$$

(Eq In)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash M(x).P \leftrightarrow M'(x).P'}$$

(Eq Par)
$$\frac{(fr, th) \vdash P \leftrightarrow P' \quad (fr, th) \vdash Q \leftrightarrow Q'}{(fr, th) \vdash P \mid Q \leftrightarrow P' \mid Q'}$$

(Eq Nil)
$$\frac{}{(fr, th) \vdash \mathbf{0} \leftrightarrow \mathbf{0}}$$

(Eq Res) (where $n \notin fr \cup fn(th)$)
$$\frac{(fr \cup \{n\}, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash (\nu n)P \leftrightarrow (\nu n)P'}$$

(Eq Repl)
$$\frac{(fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash !P \leftrightarrow !P'}$$

(Eq Match)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N' \quad (fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash [M \; is \; N]\; P \leftrightarrow [M' \; is \; N']\; P'}$$

(Eq Let)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash let\ (x, y) = M\ in\ P \leftrightarrow let\ (x, y) = M'\ in\ P'}$$

(Eq IntCase)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash P \leftrightarrow P' \quad (fr, th) \vdash Q \leftrightarrow Q'}{(fr, th) \vdash case\ M\ of\ 0 : P\ suc(x) : Q \leftrightarrow case\ M'\ of\ 0 : P'\ suc(x) : Q'}$$

(Eq Decrypt)
$$\frac{(fr, th) \vdash M \leftrightarrow M' \quad (fr, th) \vdash N \leftrightarrow N' \quad (fr, th) \vdash P \leftrightarrow P'}{(fr, th) \vdash case\ M\ of\ \{x\}_N\ in\ P \leftrightarrow case\ M'\ of\ \{x\}_{N'}\ in\ P'}$$

The core of the proof depends on a relation, $\mathcal{S}$, defined so that $P\ \mathcal{S}\ Q$ if and only if there is a frame fr, a theory th, and processes P_1, P_2, Q_1, Q_2, such that

$$P = (\nu\vec{p})(P_1 \mid P_2) \qquad Q = (\nu\vec{q})(Q_1 \mid Q_2)$$

and $(fr, th) \vdash ok$, $(fr, th) \vdash P_1 \sim_f Q_1$, and $(fr, th) \vdash P_2 \leftrightarrow Q_2$, where $\{\vec{p}\} = (fn(P_1) \cup fn(\pi_1(th))) - fr$ and $\{\vec{q}\} = (fn(Q_1) \cup fn(\pi_2(th))) - fr$. By a detailed case analysis, we may show that $\mathcal{S}$ satisfies the definition of a standard notion of bisimulation—a barbed bisimulation up to restriction and barbed equivalence. Given some auxiliary lemmas about testing equivalence, the theorem then follows easily. The construction of $\mathcal{S}$ also yields that framed bisimilarity is a sufficient condition for a strong equivalence called barbed congruence.

The converse of soundness—completeness—does not hold. The failure of completeness follows from the fact that framed bisimilarity is a sufficient condition for barbed congruence. (Barbed congruence and a fortiori framed bisimilarity are sensitive to τ steps and to branching structure, while testing equivalence is not.) In fact, as the Appendix explains, framed bisimilarity does not even seem to be a necessary condition for barbed congruence. Incompleteness may be somewhat unfortunate but, in our experience, it seems to be compatible with usefulness.

4. Examples

This section shows how bisimulations can be exploited in proofs through some small examples. These examples could not be handled by standard notions of bisimulation (like that of our earlier work [3, 4]). We have worked through further examples, including some examples with more steps. In all cases, the proofs are rather straightforward.

Throughout this section, c, K, K_1, and K_2 are distinct names, and n is any name different from c. Moreover, M, M', M'', M_1, M_2, N_1, and N_2 are closed terms; it is convenient to assume that no name occurs in them.

Example 1

As a first example, we show that the processes

$$(\nu K)\overline{c}\langle\{M\}_K\rangle$$

and

$$(\nu K)\overline{c}\langle\{M'\}_K\rangle$$

are in a framed bisimulation, so they are testing equivalent. Intuitively, this means that these processes do not reveal M and M', respectively.

For this example, we let $\mathcal{S}$ be the least relation such that:

- $(\{c, n\}, \emptyset) \vdash (\nu K)\overline{c}\langle\{M\}_K\rangle\ \mathcal{S}\ (\nu K)\overline{c}\langle\{M'\}_K\rangle$

- $(\{c, n\}, \{(\{M\}_k, \{M'\}_k)\}) \vdash \mathbf{0}\ \mathcal{S}\ \mathbf{0}$
  for all names $k \notin \{c, n\}$

Since $(\{c, n\}, \emptyset) \vdash ok$ and $(\{c, n\}, \{(\{M\}_k, \{M'\}_k)\}) \vdash ok$, $\mathcal{S}$ is a framed relation. Next we show that $\mathcal{S}$ is a framed simulation; symmetric reasoning establishes that $\mathcal{S}^{-1}$ is one too. Assuming that $(fr, th) \vdash P\ \mathcal{S}\ Q$, we need to examine the commitments of P and Q. We consider two cases, which correspond to the two clauses of the definition of $\mathcal{S}$.

- Suppose that $P = (\nu K)\overline{c}\langle\{M\}_K\rangle$ and $Q = (\nu K)\overline{c}\langle\{M'\}_K\rangle$. In this case, we have $fr = \{c, n\}$ and $th = \emptyset$. Up to renaming of the bound name K, the only commitment of P is $P \xrightarrow{\overline{c}} (\nu K)\langle\{M\}_K\rangle\mathbf{0}$. To establish that $\mathcal{S}$ is a framed simulation, we need only consider the case where K is renamed to some $k \notin fn(P) \cup fn(\pi_1(\emptyset)) \cup \{c, n\}$, that is, $k \notin \{c, n\}$. By renaming, we have $Q \xrightarrow{\overline{c}} (\nu k)\langle\{M'\}_k\rangle\mathbf{0}$. We let $th' = \{(\{M\}_k, \{M'\}_k)\}$. We have $(fr, th) \leq (fr, th')$, $(fr, th') \vdash \{M\}_k \leftrightarrow \{M'\}_k$, and $(fr, th') \vdash \mathbf{0}\ \mathcal{S}\ \mathbf{0}$ by definition of $\mathcal{S}$. Thus, Q can match P's commitment.
- Suppose that $P = \mathbf{0}$ and $Q = \mathbf{0}$. This case is trivial, since $\mathbf{0}$ has no commitments.

In short, $(\{c, n\}, \emptyset) \vdash (\nu K)\overline{c}\langle\{M\}_K\rangle\ \mathcal{S}\ (\nu K)\overline{c}\langle\{M'\}_K\rangle$, and $\mathcal{S}$ is a framed bisimulation, as desired.

Example 2

As a small variant of the first example, we consider the processes

$$(\nu K)\overline{c}\langle(\{M\}_K, \{M\}_K)\rangle$$

and

$$(\nu K)\overline{c}\langle(\{M'\}_K, \{M''\}_K)\rangle$$

When $M' = M''$, the argument of the first example works for this example too, with only trivial modifications. We define $\mathcal{S}$ as the least relation such that:

- $(\{c, n\}, \emptyset) \vdash (\nu K)\overline{c}\langle(\{M\}_K, \{M\}_K)\rangle\ \mathcal{S}\ (\nu K)\overline{c}\langle(\{M'\}_K, \{M''\}_K)\rangle$
- $(\{c, n\}, \{(\{M\}_k, \{M'\}_k), (\{M\}_k, \{M''\}_k)\}) \vdash \mathbf{0}\ \mathcal{S}\ \mathbf{0}$
  for all names $k \notin \{c, n\}$

This relation is a framed bisimulation when $M' = M''$. On the other hand, it is not a framed bisimulation when $M' \neq M''$. In fact, in that case it is not even a framed relation, because

$$(\{c, n\}, \{(\{M\}_k, \{M'\}_k), (\{M\}_k, \{M''\}_k)\}) \vdash ok$$

does not hold (because condition (2) of the definition of ok is not satisfied).

The fact that $\mathcal{S}$ is not a framed bisimulation in this case should not be a concern. It is actually necessary: the processes $(\nu K)\overline{c}\langle(\{M\}_K, \{M\}_K)\rangle$ and

$(\nu K)\overline{c}\langle(\{M'\}_K, \{M''\}_K)\rangle$ are not testing equivalent when $M' \neq M''$. The environment

$$c(z).let\ (x, y) = z\ in\ [x\ is\ y]\ \overline{c}\langle 0\rangle$$

distinguishes them. Thus, this example illustrates that two ciphertexts that cannot be decrypted can still be compared, and justifies part of the definition of framed bisimulation.

Example 3

As a further variant, we study an example with nested encryption. We consider the processes

$$(\nu K_1)(\nu K_2)\overline{c}\langle\{M_1, \{M_2\}_{K_2}\}_{K_1}\rangle.\overline{c}\langle K_1\rangle$$

and

$$(\nu K_1)(\nu K_2)\overline{c}\langle\{N_1, \{N_2\}_{K_2}\}_{K_1}\rangle.\overline{c}\langle K_1\rangle$$

Each of these processes creates two keys K_1 and K_2, sends a ciphertext, and then reveals K_1. Anyone who receives K_1 can partially decrypt the ciphertext.

In order to analyze these processes, we let $\mathcal{S}$ be the least relation such that:

- $(\{c, n\}, \emptyset) \vdash (\nu K_1)(\nu K_2)\overline{c}\langle\{M_1, \{M_2\}_{K_2}\}_{K_1}\rangle.\overline{c}\langle K_1\rangle\ \mathcal{S}$ $(\nu K_1)(\nu K_2)\overline{c}\langle\{N_1, \{N_2\}_{K_2}\}_{K_1}\rangle.\overline{c}\langle K_1\rangle$
- $(\{c, n, k_1\}, \{(\{M_2\}_{k_2}, \{N_2\}_{k_2})\}) \vdash \overline{c}\langle k_1\rangle\ \mathcal{S}\ \overline{c}\langle k_1\rangle$ for all names k_1, k_2 with $k_1 \neq k_2$ and $\{k_1, k_2\} \cap \{c, n\} = \emptyset$
- $(\{c, n, k_1\}, \{(\{M_2\}_{k_2}, \{N_2\}_{k_2})\}) \vdash \mathbf{0}\ \mathcal{S}\ \mathbf{0}$ for all names k_1, k_2 with $k_1 \neq k_2$ and $\{k_1, k_2\} \cap \{c, n\} = \emptyset$

Note how, between the first and the second clauses, the frame has been enlarged with k_1, although the processes considered in the second clause have not yet sent k_1; this simplifies the construction of $\mathcal{S}$ and is permitted by the definitions of Section 3. The assumptions guarantee that

$$(\{c, n, k_1\}, \{(\{M_2\}_{k_2}, \{N_2\}_{k_2})\}) \vdash ok$$

and hence that $\mathcal{S}$ is a framed relation. Moreover, $\mathcal{S}$ is a framed bisimulation if and only if the following condition holds:

$$(\{c, n, k_1\}, \{(\{M_2\}_{k_2}, \{N_2\}_{k_2})\}) \vdash \{M_1, \{M_2\}_{k_2}\}_{k_1} \leftrightarrow \{N_1, \{N_2\}_{k_2}\}_{k_1}$$

In turn, this condition holds if and only if $M_1 = N_1$ (because no name occurs in M_1 or N_1). Intuitively, the equality $M_1 = N_1$ becomes necessary only when the two processes send the key k_1, since M_1 and N_1 are not visible in the first message. Our definition of $\leq$ guarantees that the necessity of $M_1 = N_1$ is propagated correctly.

Example 4

Yet another variant illustrates the use of replication in combination with restriction. We consider the processes

$$!(\nu K)\overline{c}\langle\{M\}_K\rangle$$

and

$$(\nu K)!(\nu K')\overline{c}\langle\{K', M\}_K\rangle$$

The former process repeatedly creates keys and sends a fixed message M encrypted under these keys. The latter process creates a fixed key, then repeatedly creates names and sends them encrypted under the key along with the fixed message M. These names serve to guarantee that all the messages are different although they all contain M and rely on the same key.

For this example, we let $\mathcal{S}$ be the least relation such that $(\mathit{fr}, \mathit{th}) \vdash P \ \mathcal{S}\ Q$ when:

- $\mathit{fr} = \{c, n\}$;
- th is any finite set of pairs $(\{M\}_{k_i}, \{k'_i, M\}_k)$, each with different names k_i and k'_i but all with the same term M and name k, with $k_i, k \notin \mathit{fr}$;
- P is the parallel composition of $!(\nu K)\overline{c}\langle\{M\}_K\rangle$ with an arbitrary number of copies of $(\nu K)\overline{c}\langle\{M\}_K\rangle$ and with an arbitrary number of copies of $\mathbf{0}$;
- Q is either the parallel composition of $(\nu K)!(\nu K')\overline{c}\langle\{K', M\}_K\rangle$ with an arbitrary number of copies of $\mathbf{0}$ if th is empty or the parallel composition of $!(\nu K')\overline{c}\langle\{K', M\}_k\rangle$ with an arbitrary number of copies of $(\nu K')\overline{c}\langle\{K', M\}_k\rangle$ and with an arbitrary number of copies of $\mathbf{0}$ if th contains a pair $(\{M\}_{k_i}, \{k'_i, M\}_k)$.

These assumptions imply that $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$ and hence that $\mathcal{S}$ is a framed relation. Moreover, if $(\mathit{fr}, \mathit{th}) \vdash P\ \mathcal{S}\ Q$, then the only commitment of P is an output on c yielding a concretion $(\nu k_i)\langle\{M\}_{k_i}\rangle P'$ where P' has the same form as P. Similarly, the only commitment of Q is an output on c yielding the concretion $(\nu k, k'_i)\langle\{k'_i, M\}_k\rangle Q'$ when th is empty and the concretion $(\nu k'_i)\langle\{k'_i, M\}_k\rangle Q'$ when th contains a pair $(\{M\}_{k_i}, \{k'_i, M\}_k)$; in both cases, Q' is the parallel composition of $!(\nu K')\overline{c}\langle\{K', M\}_k\rangle$ with an arbitrary number of copies of $(\nu K')\overline{c}\langle\{K', M\}_k\rangle$ and with an arbitrary number of copies of $\mathbf{0}$. We can take k_i and k'_i to be fresh names, and also k if th is empty. Letting $\mathit{th}' = \mathit{th} \cup \{(\{M\}_{k_i}, \{k'_i, M\}_k)\}$, we obtain $(\mathit{fr}, \mathit{th}) \leq (\mathit{fr}, \mathit{th}')$, $(\mathit{fr}, \mathit{th}') \vdash \{M\}_{k_i} \leftrightarrow \{k'_i, M\}_k$, and $(\mathit{fr}, \mathit{th}') \vdash P'\ \mathcal{S}\ Q'$. It follows that $\mathcal{S}$ is a framed bisimulation.

Example 5

While all the examples above concern the secrecy of certain outputs, this one concerns the impossibility of certain inputs. We consider the processes

$$(\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).[x\ \mathit{is}\ K]\ \overline{c}\langle\{0\}_K\rangle$$

and

$$(\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).\mathbf{0}$$

The former process creates a key K, sends $\{0\}_K$, listens for an input, and if it receives K then it sends $\{0\}_K$ again. However, we would expect that K will never arrive as an input to this process, since the process never discloses K (but only $\{0\}_K$, from which K itself cannot be deduced). Therefore, we would expect this process to be equivalent to the latter process, which simply stops upon receipt of a message.

For this example, we let $\mathcal{S}$ be the least relation such that:

- $(\{c,n\},\emptyset) \vdash ((\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).[x\ is\ K]\ \overline{c}\langle\{0\}_K\rangle)\ \mathcal{S}$
  $((\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).\mathbf{0})$
- $(\{c,n\},\{(\{0\}_k,\{0\}_k)\}) \vdash (c(x).[x\ is\ k]\ \overline{c}\langle\{0\}_k\rangle)\ \mathcal{S}\ (c(x).\mathbf{0})$
  for all names k with $k \notin \{c,n\}$
- $(\{c,n,\vec{m}\},\{(\{0\}_k,\{0\}_k)\}) \vdash ([N\ is\ k]\ \overline{c}\langle\{0\}_k\rangle)\ \mathcal{S}\ \mathbf{0}$
  for all names k with $k \notin \{c,n\}$, for all sets $\{\vec{m}\}$ disjoint from $\{c,n,k\}$, and all closed terms N and N' with $(\{c,n,\vec{m}\},\{(\{0\}_k,\{0\}_k)\}) \vdash N \leftrightarrow N'$ (We are not assuming that no names occur in the closed terms N and N'.)

Since $(\{c,n,\vec{m}\},\{(\{0\}_k,\{0\}_k)\}) \vdash N \leftrightarrow N'$ and $k \notin \{c,n,\vec{m}\}$, the term N is not k, so $[N\ is\ k]\ \overline{c}\langle\{0\}_k\rangle \xrightarrow{\alpha} A$ is not true for any α and A. In other words, the process $[N\ is\ k]\ \overline{c}\langle\{0\}_k\rangle$ is stuck. It follows easily that $\mathcal{S}$ is a framed bisimulation.

Example 6

This example is a variant of Example 5; it illustrates the flexibility of our ordering on frame-theory pairs. We consider the processes

$$(\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).(\overline{c}\langle K\rangle \mid [x\ is\ K]\ \overline{c}\langle\{0\}_K\rangle)$$

and

$$(\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).\overline{c}\langle K\rangle$$

Compared to the corresponding processes of Example 5, these have an additional message where the key K is disclosed. This message is emitted only after the input x is received, so it does not affect the possible values of this input.

We let $\mathcal{S}$ be the least relation such that:

- $(\{c,n\},\emptyset) \vdash ((\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).(\overline{c}\langle K\rangle \mid [x\ is\ K]\ \overline{c}\langle\{0\}_K\rangle))\ \mathcal{S}$
  $((\nu K)\overline{c}\langle\{0\}_K\rangle.c(x).\overline{c}\langle K\rangle)$
- $(\{c,n\},\{(\{0\}_k,\{0\}_k)\}) \vdash c(x).(\overline{c}\langle k\rangle \mid [x\ is\ k]\ \overline{c}\langle\{0\}_k\rangle)\ \mathcal{S}\ c(x).\overline{c}\langle k\rangle$
  for all names k with $k \notin \{c,n\}$
- $(\{c,n,\vec{m}\},\{(\{0\}_k,\{0\}_k)\}) \vdash (\overline{c}\langle k\rangle \mid [N\ is\ k]\ \overline{c}\langle\{0\}_k\rangle)\ \mathcal{S}\ \overline{c}\langle k\rangle$
  for all names k with $k \notin \{c,n\}$, for all sets $\{\vec{m}\}$ disjoint from $\{c,n,k\}$, and all closed terms N and N' with $(\{c,n,\vec{m}\},\{(\{0\}_k,\{0\}_k)\}) \vdash N \leftrightarrow N'$

- $(\{c, n, \vec{m}, k\}, \emptyset) \vdash (\mathbf{0} \mid [N \; is \; k] \, \overline{c}\langle\{0\}_k\rangle) \; \mathcal{S} \; \mathbf{0}$
  for all names k with $k \notin \{c, n\}$, for all sets $\{\vec{m}\}$ disjoint from $\{c, n, k\}$, and all closed terms N and N' with $(\{c, n, \vec{m}\}, \{(\{0\}_k, \{0\}_k)\}) \vdash N \leftrightarrow N'$

As in Example 5, $\mathcal{S}$ is a framed bisimulation because N cannot equal k. Note how the first three clauses of this definition have the pair $\{(\{0\}_k, \{0\}_k)\}$ as part of a theory, while the last clause has k in a frame; this shift from theory to frame is essential.

Example 7

In cryptographic protocols, some keys are generated by consulting sources of randomness, but it is also common to generate keys by applying one-way functions to other keys. (Whereas many one-way functions are quite efficient, randomness and key agreement can be relatively expensive [22].) As a final example, we consider a simple protocol transformation inspired by a common method for generating keys. We compare the process

$$(\nu K_1)(\nu K_2)\overline{c}\langle\{M_1\}_{K_1}\rangle.\overline{c}\langle\{M_2\}_{K_2}\rangle$$

which generates and uses two keys, with the process

$$(\nu K)\overline{c}\langle\{N_1\}_{\{0\}_K}\rangle.\overline{c}\langle\{N_2\}_{\{1\}_K}\rangle$$

which generates the master key K and then uses the derived keys $\{0\}_K$ and $\{1\}_K$.

In order to show that these processes are testing equivalent, we construct once more a framed bisimulation. We let $\mathcal{S}$ be the least relation such that:

- $(\{c, n\}, \emptyset) \vdash (\nu K_1)(\nu K_2)\overline{c}\langle\{M_1\}_{K_1}\rangle.\overline{c}\langle\{M_2\}_{K_2}\rangle \; \mathcal{S}$
  $(\nu K)\overline{c}\langle\{N_1\}_{\{0\}_K}\rangle.\overline{c}\langle\{N_2\}_{\{1\}_K}\rangle$
- $(\{c, n\}, \{(\{M_1\}_{k_1}, \{N_1\}_{\{0\}_k})\}) \vdash (\nu K_2)\overline{c}\langle\{M_2\}_{K_2}\rangle \; \mathcal{S} \; \overline{c}\langle\{N_2\}_{\{1\}_k}\rangle$
  for all names k and k_1 with $\{k, k_1\} \cap \{c, n\} = \emptyset$
- $(\{c, n\}, \{(\{M_1\}_{k_1}, \{N_1\}_{\{0\}_k}), (\{M_2\}_{k_2}, \{N_2\}_{\{1\}_k})\}) \vdash \mathbf{0} \; \mathcal{S} \; \mathbf{0}$
  for all names k and k_1 with $\{k, k_1\} \cap \{c, n\} = \emptyset$
  and all names k_2 with $k_2 \notin \{c, n, k_1\}$

It is somewhat laborious but not difficult to check that $\mathcal{S}$ is a framed bisimulation, much as in the examples above.

5. Related Work

Park [17] first suggested the bisimulation proof technique, in the context of Milner's CCS. After Park's work, bisimulation became a cornerstone of the theory of CCS [15]. Milner, Parrow, and Walker [16] extensively studied a variety of forms of bisimulation for the pi calculus, their generalization of CCS with name-passing and mobile restrictions. Our definition of framed bisimulation generalizes (and relaxes) the definition of strong bisimulation from earlier work on the spi calculus [3, 4]. We can show that if processes P and Q are strongly bisimilar, then, for all frames fr, $(fr, \emptyset) \vdash P \sim_f Q$. The converse implication fails.

According to most other definitions, a bisimulation is a set of pairs of processes. According to our definition, a framed bisimulation is a set of quadruples consisting of a pair of processes grouped with a frame and a theory. According to a definition of Pierce and Sangiorgi [19] for a typed pi calculus, a bisimulation is a set of pairs of processes indexed by a type assumption that binds types to channel names. Our use of a frame is a little like their use of typing assumptions, in that both a frame and a typing assumption delimit the channels on which the environment may observe the processes in the bisimulation. On the other hand, the spi calculus is untyped, and our use of a theory to represent compound terms possessed but not decomposable by the environment seems to be new.

There are also parallels with the work of Pitts and Stark [20] on the ν-calculus, a simply-typed λ-calculus enriched with dynamic allocation of names. Pitts and Stark define a logical relation on programs, parameterized by a partial bijection on the names free in related programs. Their logical relation is sound for proving observational equivalence; it is incomplete but more generous than the usual notion of applicative bisimilarity for the ν-calculus. A logical relation is one in which two abstractions are related if and only if they send related arguments to related results. Given the clause for inputs in the definition of framed bisimulation, which requires the bodies of two abstractions $(x)P'$ and $(x)Q'$ to be related on all related terms M and N, we may say that the relations $(fr, th) \vdash M \leftrightarrow N$ and $(fr, th) \vdash P \sim_f Q$ form a parametric logical relation on the terms and processes of the spi calculus. Like Pitts and Stark's logical relation, our logical relation is sound for proving testing equivalence. Further, it is incomplete but more generous than the usual notion of strong bisimilarity; it has parameters (the frame and the theory) that serve to identify certain processes that are distinguished by the usual relation of strong bisimilarity. However, the analogy with Pitts and Stark's work is not perfect; in particular, their use of partial bijections on names is different from our use of frames and theories.

In the last few years, several methods for analyzing cryptographic protocols have been developed within action-based or state-based models (see for example [5, 9, 10, 11, 12, 13, 14, 18, 21]). Some of these models are presented as process algebras, others in logical forms. Often, the analysis of a protocol requires defining a particular attacker (an environment) for the protocol; recently, there has been promising progress towards automating the construction of this attacker. Bisimulation techniques do appear in the security literature (as in the work of Focardi and Gorrieri [8]), but rarely, and without special tailoring to cryptographic applications.

6. Conclusions

When reasoning about a cryptographic protocol, we must take into account the knowledge of the environment with which the protocol interacts. In our definition of bisimulation, this knowledge is represented precisely as a set of names that the environment has obtained, and as a set of pairs of ciphertexts that the environment has received but cannot distinguish. This precise representation of the knowledge of the environment is the basis for an effective and sound proof technique. Using this technique, we can construct proofs for small but subtle cryptographic protocols. The proofs are fairly concise and do not require much creativity. Therefore, although we have not yet attempted to mechanize our proofs, we hope that such a mechanization is possible, and that it may enable the automatic verification of substantial examples. Recent, encouraging work has started to address the obstacles to a mechanization [7].

Acknowledgements

Discussions with Davide Sangiorgi were helpful at the start of this work. Andy Pitts made useful comments.

Gordon held a Royal Society University Research Fellowship at the University of Cambridge Computer Laboratory for most of the time we worked on this paper.

Appendix A. Soundness of Framed Bisimulation

This appendix shows that framed bisimulation is a sufficient condition for testing equivalence, as announced in Section 3. This soundness result requires a number of auxiliary notations, definitions, and lemmas. One of the central auxiliary lemmas is that framed bisimulation is also a sufficient condition for a strong equivalence called barbed congruence.

Throughout this Appendix, we rely on definitions that appear in earlier work. In Appendix A.1, we review the rules defining the reduction relation, $P > Q$, and the commitment relation, $P \xrightarrow{\alpha} A$. For the sake of brevity, we omit the definitions of standard π-calculus notions of structural equivalence $\equiv$, barbed bisimulation, barbed equivalence $\stackrel{\bullet}{\sim}$, barbed congruence, and the notion of a barbed bisimulation up to restriction and $\stackrel{\bullet}{\sim}$. These definitions appear in earlier papers [3].

Appendix A.1 Commitment (Review)

We extend the restriction and composition operators to arbitrary agents, as follows. For an abstraction $(x)P$, we let $(\nu m)(x)P = (x)(\nu m)P$ and let $R \mid (x)P = (x)(R \mid P)$, assuming that $x \notin fv(R)$. For a concretion $(\nu\vec{n})\langle M\rangle Q$, we let $(\nu m)(\nu\vec{n})\langle M\rangle Q = (\nu m, \vec{n})\langle M\rangle Q$ if $m \in fn(M)$ and $= (\nu\vec{n})\langle M\rangle(\nu m)Q$ otherwise, and let $R \mid (\nu\vec{n})\langle M\rangle Q = (\nu\vec{n})\langle M\rangle(R \mid Q)$, in all cases assuming that $m \notin \{\vec{n}\}$ and that $\{\vec{n}\} \cap fn(R) = \emptyset$. We define the dual composition $A \mid R$ symmetrically.

If F is the abstraction $(x)P$ and C is the concretion $(\nu\vec{n})\langle M\rangle Q$, and $\{\vec{n}\} \cap fn(P) = \emptyset$, we define the *interactions* $F@C$ and $C@F$ to be the processes given by:

$$F@C \triangleq (\nu\vec{n})(P[M/x] \mid Q) \qquad C@F \triangleq (\nu\vec{n})(Q \mid P[M/x])$$

We define a *reduction* relation $>$ on closed processes by the axioms:

(Red Repl)	$!P$	$>$	$P \mid !P$
(Red Match)	$[M\ is\ M]\ P$	$>$	P
(Red Let)	$let\ (x,y) = (M,N)\ in\ P$	$>$	$P[M/x][N/y]$
(Red Zero)	$case\ 0\ of\ 0 : P\ suc(x) : Q$	$>$	P
(Red Suc)	$case\ suc(M)\ of\ 0 : P\ suc(x) : Q$	$>$	$Q[M/x]$
(Red Decrypt)	$case\ \{M\}_N\ of\ \{x\}_N\ in\ P$	$>$	$P[M/x]$

Commitment is defined inductively by the following axioms and rules:

(Comm In) $m(x).P \xrightarrow{m} (x)P$ (Comm Out) $\overline{m}\langle M\rangle.P \xrightarrow{\overline{m}} (\nu)\langle M\rangle P$

(Comm Inter 1)
$$\frac{P \xrightarrow{m} F \quad Q \xrightarrow{\overline{m}} C}{P \mid Q \xrightarrow{\tau} F@C}$$

(Comm Inter 2)
$$\frac{P \xrightarrow{\overline{m}} C \quad Q \xrightarrow{m} F}{P \mid Q \xrightarrow{\tau} C@F}$$

(Comm Red)
$$\frac{P > Q \quad Q \xrightarrow{\alpha} A}{P \xrightarrow{\alpha} A}$$

(Comm Par 1)
$$\frac{P \xrightarrow{\alpha} A}{P \mid Q \xrightarrow{\alpha} A \mid Q}$$

(Comm Par 2)
$$\frac{Q \xrightarrow{\alpha} A}{P \mid Q \xrightarrow{\alpha} P \mid A}$$

(Comm Res)
$$\frac{P \xrightarrow{\alpha} A \quad \alpha \notin \{m, \overline{m}\}}{(\nu m)P \xrightarrow{\alpha} (\nu m)A}$$

Appendix A.2 Auxiliary Lemmas about Commitment and Testing Equivalence

LEMMA 1. *Consider two closed processes P and Q. Suppose that, for every name n, every barb $\beta \in \{n, \overline{n}\}$, and every R such that $fn(R) \subseteq fn(P) \cup fn(Q) \cup \{n\}$, if*

$$P \mid R \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A, then

$$Q \mid R \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B. Then, for every barb β and every R, if

$$P \mid R \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A, then

$$Q \mid R \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B.

PROOF. For all P and β, let $P \Downarrow \beta$ if and only if $P \xrightarrow{\tau}{}^{*} P'$ and $P' \xrightarrow{\beta} A$ for some P' and A. In earlier work [3], we established the following facts: (1) that if $P \equiv Q$ and $Q \Downarrow \beta$ then $P \Downarrow \beta$; and (2) that if $\beta \notin \{n, \overline{n}\}$ then $P \Downarrow \beta$ if and only if $(\nu n)P \Downarrow \beta$.

Suppose that $P \mid R \Downarrow \beta$. Let n be the name such that $\beta \in \{n, \overline{n}\}$. Let $m_0, \ldots, m_k$ be the names that appear free in R but not in P or Q, and that are different from n. By (2), we get that $(\nu m_0)\cdots(\nu m_k)(P \mid R) \Downarrow \beta$. Since $m_0, \ldots, m_k$ do not appear free in P, we have that $(\nu m_0)\cdots(\nu m_k)(P \mid R) \equiv P \mid (\nu m_0)\cdots(\nu m_k)R$. By (1), we get that $P \mid (\nu m_0)\cdots(\nu m_k)R \Downarrow \beta$. By assumption, this implies that $Q \mid (\nu m_0)\cdots(\nu m_k)R \Downarrow \beta$. Since $m_0, \ldots, m_k$ do not appear free in Q, we have that $Q \mid (\nu m_0)\cdots(\nu m_k)R \equiv (\nu m_0)\cdots(\nu m_k)(Q \mid R)$. By (1), we get that $(\nu m_0)\cdots(\nu m_k)(Q \mid R) \Downarrow \beta$. By (2), we conclude that $Q \mid R \Downarrow \beta$. □

COROLLARY 1. *Consider two closed processes P and Q. Suppose that, for every name n, every barb $\beta \in \{n, \overline{n}\}$, and every R such that $fn(R) \subseteq fn(P) \cup fn(Q) \cup \{n\}$,*

$$P \mid R \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A if and only if

$$Q \mid R \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B. Then $P \simeq Q$.

PROOF. This is a direct consequence of Lemma 1. □

PROPOSITION 2. *Consider two closed processes P and Q, and a barb β. If $P \stackrel{\bullet}{\sim} Q$ and*

$$P \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A, then

$$Q \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B.

PROOF. The proof is by induction on the number of steps in $P \xrightarrow{\tau}{}^{*} P'$. If that number is 0, then $P \xrightarrow{\beta} A$ and, since $\stackrel{\bullet}{\sim}$ is a barbed bisimulation, it follows that $Q \xrightarrow{\beta} B$ for some B. If that number is not 0, on the other hand, we have $P \xrightarrow{\tau} P'' \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$ for some P''. Since $\stackrel{\bullet}{\sim}$ is a barbed bisimulation, there exists Q'' such that $Q \xrightarrow{\tau} Q'$ and $P'' \equiv\stackrel{\bullet}{\sim}\equiv Q''$. Since structural equivalence implies barbed equivalence and since barbed equivalence is transitive [3], we have $P'' \stackrel{\bullet}{\sim} Q''$. By induction hypothesis, there exist Q' and B such that $Q \xrightarrow{\tau} Q'' \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$. □

We use Proposition 2 in the proof of our main soundness result, Theorem 1.

Appendix A.3 Auxiliary Notations and Definitions

This section obtains an alternative formulation of framed bisimulation that we use later on in the appendix.

We add the following two rules for abstractions and concretions:

$$\text{(Eq Abstraction)}\quad \frac{(\mathit{fr}, \mathit{th}) \vdash P \leftrightarrow P'}{(\mathit{fr}, \mathit{th}) \vdash (x)P \leftrightarrow (x)P'}$$

$$\text{(Eq Concretion) (where } \{\vec{n}\} \cap (\mathit{fr} \cup \mathit{fn}(\mathit{th})) = \emptyset)\quad \frac{(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash M \leftrightarrow M' \quad (\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash P \leftrightarrow P'}{(\mathit{fr}, \mathit{th}) \vdash (\nu\vec{n})\langle M\rangle P \leftrightarrow (\nu\vec{n})\langle M'\rangle P'}$$

If $\mathcal{R}$ is a framed relation, we define the predicates $(\mathit{fr}, \mathit{th}) \vdash (x)P \,\overline{\mathcal{R}}\, (x)Q$ and $(\mathit{fr}, \mathit{th}) \vdash \langle M\rangle P \,\overline{\mathcal{R}}\, \langle N\rangle Q$ as follows:

- $(\mathit{fr}, \mathit{th}) \vdash (x)P \,\overline{\mathcal{R}}\, (x)Q$ if and only if for all closed M and N, $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N$ implies $(\mathit{fr}, \mathit{th}) \vdash P[M/x] \,\mathcal{R}\, Q[N/x]$;
- $(\mathit{fr}, \mathit{th}) \vdash \langle M\rangle P \,\overline{\mathcal{R}}\, \langle N\rangle Q$ if and only if there is a frame fr' and a theory th' such that $(\mathit{fr}, \mathit{th}) \leq (\mathit{fr}', \mathit{th}')$, $(\mathit{fr}', \mathit{th}') \vdash M \leftrightarrow N$, and $(\mathit{fr}', \mathit{th}') \vdash P \,\mathcal{R}\, Q$.

With these definitions, we obtain that a framed simulation is a framed relation $\mathcal{S}$ such that $(\mathit{fr}, \mathit{th}) \vdash P \,\mathcal{S}\, Q$ implies:

- $P \xrightarrow{\tau} P'$ implies that there is Q' with $Q \xrightarrow{\tau} Q'$ and $(\mathit{fr}, \mathit{th}) \vdash P' \,\mathcal{S}\, Q'$;
- $P \xrightarrow{c} F$ and $c \in \mathit{fr}$ imply that there is G with $Q \xrightarrow{c} G$ such that $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash F \,\overline{\mathcal{S}}\, G$ for all sets $\{\vec{n}\}$ disjoint from $\mathit{fn}(P) \cup \mathit{fn}(Q) \cup \mathit{fr} \cup \mathit{fn}(\mathit{th})$;
- $P \xrightarrow{\overline{c}} (\nu\vec{m})\langle M\rangle P'$, $c \in \mathit{fr}$, and the set $\{\vec{m}\}$ is disjoint from $\mathit{fn}(P) \cup \mathit{fn}(\pi_1(\mathit{th})) \cup \mathit{fr}$ imply that there is $D = (\nu\vec{n})\langle N\rangle Q'$ such that $Q \xrightarrow{\overline{c}} D$, the set $\{\vec{n}\}$ is disjoint from $\mathit{fn}(Q) \cup \mathit{fn}(\pi_2(\mathit{th})) \cup \mathit{fr}$, and $(\mathit{fr}, \mathit{th}) \vdash \langle M\rangle P' \,\overline{\mathcal{S}}\, \langle N\rangle Q'$.

Appendix A.4 Auxiliary Lemmas about Framed Bisimulations

In Appendix A.5, we prove a main lemma, Lemma 14, that leads, in Appendix A.6, to a proof of our soundness results. Here we prove a series of lemmas used to establish Lemma 14.

LEMMA 2.

- *If $\mathit{fn}(M) \subseteq \mathit{fr}$ then $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow M$.*
- *If $\mathit{fn}(P) \subseteq \mathit{fr}$ then $(\mathit{fr}, \mathit{th}) \vdash P \leftrightarrow P$.*

PROOF. By induction on the structure of M and P. □

LEMMA 3.

(1) $(fr, th) \vdash M \leftrightarrow N$ *implies* $fn(M) \subseteq fr \cup fn(\pi_1(th))$ *and* $fn(N) \subseteq fr \cup fn(\pi_2(th))$.

(2) $(fr, th) \vdash A \leftrightarrow B$ *implies* $fn(A) \subseteq fr \cup fn(\pi_1(th))$ *and* $fn(B) \subseteq fr \cup fn(\pi_2(th))$.

PROOF. By inductions on the derivations of $(fr, th) \vdash M \leftrightarrow N$ and $(fr, th) \vdash A \leftrightarrow B$. $\square$

LEMMA 4. *Suppose* $(fr, th) \vdash ok$ *and* $(fr, th) \vdash M \leftrightarrow N$.

(1) *If* $(fr, th) \vdash M \leftrightarrow N'$ *then* $N = N'$.

(2) *If* $(fr, th) \vdash M' \leftrightarrow N$ *then* $M = M'$.

PROOF. By induction on the derivation of $(fr, th) \vdash M \leftrightarrow N$. We consider the proof of part (1) in detail; part (2) follows by a symmetric argument.

(Eq Theory) In this case, $(M, N) \in th$. Suppose that $(fr, th) \vdash M \leftrightarrow N'$. Given $(fr, th) \vdash ok$, $(M, N) \in th$ implies that M is closed and there are terms M_1 and M_2 such that $M = \{M_1\}_{M_2}$ and there is no N_2 such that $(fr, th) \vdash M_2 \leftrightarrow N_2$. By inspection, only (Eq Theory) or (Eq Encrypt) can derive $(fr, th) \vdash \{M_1\}_{M_2} \leftrightarrow N'$. But rule (Eq Encrypt) cannot derive this judgment. If it did, there would need to be terms N_1' and N_2' such that $N' = \{N_1'\}_{N_2'}$ and $(fr, th) \vdash M_1 \leftrightarrow N_1'$ and $(fr, th) \vdash M_2 \leftrightarrow N_2'$. But this contradicts the fact that there is no N_2 such that $(fr, th) \vdash M_2 \leftrightarrow N_2$. Now, since (Eq Theory) must derive $(fr, th) \vdash M \leftrightarrow N'$, it follows that $(M, N') \in th$. Given $(fr, th) \vdash ok$, $(M, N) \in th$ and $(M, N') \in th$ imply that $N = N'$.

(Eq Encrypt) In this case, $M = \{M_1\}_{M_2}$, $N = \{N_1\}_{N_2}$, $(fr, th) \vdash M_1 \leftrightarrow N_1$, and $(fr, th) \vdash M_2 \leftrightarrow N_2$. Given $(fr, th) \vdash ok$, $(fr, th) \vdash M_2 \leftrightarrow N_2$ implies there can be no M' such that $(M, M') \in th$. Therefore $(fr, th) \vdash M \leftrightarrow N'$ can only be derived by (Eq Encrypt). Therefore $N' = \{N_1'\}_{N_2'}$ with $(fr, th) \vdash M_1 \leftrightarrow N_1'$ and $(fr, th) \vdash M_2 \leftrightarrow N_2'$. By induction hypothesis, $(fr, th) \vdash ok$, $(fr, th) \vdash M_1 \leftrightarrow N_1$, and $(fr, th) \vdash M_1 \leftrightarrow N_1'$ imply $N_1 = N_1'$. By induction hypothesis, $(fr, th) \vdash ok$, $(fr, th) \vdash M_2 \leftrightarrow N_2$, and $(fr, th) \vdash M_2 \leftrightarrow N_2'$ imply $N_2 = N_2'$. Hence, we have $N = \{N_1\}_{N_2} = \{N_1'\}_{N_2'} = N'$.

The cases for all the other rules for deriving $(fr, th) \vdash M \leftrightarrow N$ are simpler, as it is clear that the same rule must have been used to derive $(fr, th) \vdash M \leftrightarrow N'$. We omit these cases. $\square$

LEMMA 5. *Suppose* $n \notin fr$.

(1) *If* $(fr, th) \vdash M \leftrightarrow N$ *then* $(fr \cup \{n\}, th) \vdash M \leftrightarrow N$.

(2) *If* $(fr, th) \vdash A \leftrightarrow B$ *then* $(fr \cup \{n\}, th) \vdash A \leftrightarrow B$.

PROOF. By induction on the derivations of $(fr, th) \vdash M \leftrightarrow N$ and $(fr, th) \vdash A \leftrightarrow B$, with a renaming in the case of restriction. $\square$

LEMMA 6.
(1) *If* $(fr \cup \{n\}, th) \vdash M \leftrightarrow N$ *and* $n \notin fn(M) \cup fn(N)$ *then* $(fr, th) \vdash M \leftrightarrow N$.
(2) *If* $(fr \cup \{n\}, th) \vdash A \leftrightarrow B$ *and* $n \notin fn(A) \cup fn(B)$ *then* $(fr, th) \vdash A \leftrightarrow B$.

PROOF. By inductions on the derivations of $(fr \cup \{n\}, th) \vdash M \leftrightarrow N$ and $(fr \cup \{n\}, th) \vdash A \leftrightarrow B$. $\square$

LEMMA 7. *Suppose that* $(fr, th) \vdash M' \leftrightarrow N'$.
(1) *If* $(fr, th) \vdash M \leftrightarrow N$ *then* $(fr, th) \vdash M[M'/x] \leftrightarrow N[N'/x]$.
(2) *If* $(fr, th) \vdash A \leftrightarrow B$ *then* $(fr, th) \vdash A[M'/x] \leftrightarrow B[N'/x]$.

PROOF. By inductions on the derivations of $(fr, th) \vdash M \leftrightarrow N$ and $(fr, th) \vdash A \leftrightarrow B$ using Lemma 5 in the case of restriction. $\square$

LEMMA 8. *Let F and G be abstractions and let C and D be concretions. If* $(fr, th) \vdash F \leftrightarrow G$ *and* $(fr, th) \vdash C \leftrightarrow D$ *then* $(fr, th) \vdash F@C \leftrightarrow G@D$ *and* $(fr, th) \vdash C@F \leftrightarrow D@G$.

PROOF. Since only (Eq Abstraction) may derive $(fr, th) \vdash F \leftrightarrow G$ we have $F = (x)P_1$ and $G = (x)Q_1$ with $(fr, th) \vdash P_1 \leftrightarrow Q_1$. Since only (Eq Concretion) may derive $(fr, th) \vdash C \leftrightarrow D$ we have $C = (\nu\vec{n})\langle M\rangle P_2$ and $D = (\nu\vec{n})\langle N\rangle Q_2$ with $(fr \cup \{\vec{n}\}, th) \vdash M \leftrightarrow N$, $(fr \cup \{\vec{n}\}, th) \vdash P_2 \leftrightarrow Q_2$, and $\{\vec{n}\} \cap (fr \cup fn(th)) = \emptyset$. By Lemma 3, $(fr, th) \vdash P_1 \leftrightarrow Q_1$ implies that $fn(P_1) \cup fn(Q_1) \subseteq fr \cup fn(th)$. Therefore $\{\vec{n}\} \cap (fn(P_1) \cup fn(Q_1)) = \emptyset$ and we have:

$$\begin{aligned} F@C &= (\nu\vec{n})(P_1[M/x] \mid P_2) \\ G@D &= (\nu\vec{n})(Q_1[N/x] \mid Q_2) \end{aligned}$$

By Lemma 5, $(fr, th) \vdash P_1 \leftrightarrow Q_1$ and $\{\vec{n}\} \cap fr = \emptyset$ imply $(fr \cup \{\vec{n}\}, th) \vdash P_1 \leftrightarrow Q_1$. By Lemma 7(2), this and $(fr \cup \{\vec{n}\}, th) \vdash M \leftrightarrow N$ imply $(fr \cup \{\vec{n}\}, th) \vdash P_1[M/x] \leftrightarrow Q_1[N/x]$. By (Eq Par), this and $(fr \cup \{\vec{n}\}, th) \vdash P_2 \leftrightarrow Q_2$ imply $(fr \cup \{\vec{n}\}, th) \vdash P_1[M/x] \mid P_2 \leftrightarrow Q_1[N/x] \mid Q_2$. By (Eq Res), this and $\{\vec{n}\} \cap (fr \cup fn(th)) = \emptyset$ imply $(fr, th) \vdash (\nu\vec{n})(P_1[M/x] \mid P_2) \leftrightarrow (\nu\vec{n})(Q_1[N/x] \mid Q_2)$, that is, $(fr, th) \vdash F@C \leftrightarrow G@D$. A symmetric argument establishes that $(fr, th) \vdash C@F \leftrightarrow D@G$. $\square$

LEMMA 9. *If* $(fr, th) \vdash A \leftrightarrow B$ *and* $(fr, th) \vdash P \leftrightarrow Q$ *then* $(fr, th) \vdash A \mid P \leftrightarrow B \mid Q$ *and* $(fr, th) \vdash P \mid A \leftrightarrow Q \mid B$.

PROOF. If A and B are processes, the result follows at once using (Eq Par). Otherwise, $(fr, th) \vdash A \leftrightarrow B$ is derived via one of the following rules.

(Eq Abstraction) In this case, $A = (x)P'$ and $B = (x)Q'$ with $(fr, th) \vdash P' \leftrightarrow Q'$. Since the variable x is bound, we may assume that $x \notin fv(P) \cup fv(Q)$. Hence, since A and B are abstractions, we have $A \mid P = (x)(P' \mid P)$ and $B \mid Q = (x)(Q' \mid Q)$. By (Eq Par), $(fr, th) \vdash P' \leftrightarrow Q'$ and $(fr, th) \vdash P \leftrightarrow Q$ imply $(fr, th) \vdash P' \mid P \leftrightarrow Q' \mid Q$. By (Eq Abstraction), this implies $(fr, th) \vdash (x)(P' \mid P) \leftrightarrow (x)(Q' \mid Q)$, that is, $(fr, th) \vdash A \mid P \leftrightarrow B \mid Q$. A symmetric argument yields $(fr, th) \vdash P \mid A \leftrightarrow Q \mid B$.

(Eq Concretion) In this case $A = (\nu\vec{n})\langle M\rangle P'$ and $B = (\nu\vec{n})\langle N\rangle Q'$ with $(fr \cup \{\vec{n}\}, th) \vdash M \leftrightarrow N$, $(fr \cup \{\vec{n}\}, th) \vdash P' \leftrightarrow Q'$, and $\{\vec{n}\} \cap (fr \cup fn(th)) = \emptyset$. By Lemma 3, $(fr, th) \vdash P \leftrightarrow Q$ implies that $fn(P) \cup fn(Q) \subseteq fr \cup fn(th)$. Therefore $\{\vec{n}\} \cap (fn(P) \cup fn(Q)) = \emptyset$. Hence, since A and B are concretions, we have $A \mid P = (\nu\vec{n})\langle M\rangle(P' \mid P)$ and $B \mid Q = (\nu\vec{n})\langle N\rangle(Q' \mid Q)$. By Lemma 5, $(fr, th) \vdash P \leftrightarrow Q$ and $\{\vec{n}\} \cap fr = \emptyset$ imply $(fr \cup \{\vec{n}\}, th) \vdash P \leftrightarrow Q$. By (Eq Par), this and $(fr \cup \{\vec{n}\}, th) \vdash P' \leftrightarrow Q'$ imply $(fr \cup \{\vec{n}\}, th) \vdash P' \mid P \leftrightarrow Q' \mid Q$. By (Eq Concretion), this, $(fr \cup \{\vec{n}\}, th) \vdash M \leftrightarrow N$ and $\{\vec{n}\} \cap (fr \cup fn(th)) = \emptyset$ imply $(fr, th) \vdash (\nu\vec{n})\langle M\rangle(P' \mid P) \leftrightarrow (\nu\vec{n})\langle N\rangle(Q' \mid Q)$, that is, $(fr, th) \vdash A \mid P \leftrightarrow B \mid Q$. A symmetric argument yields $(fr, th) \vdash P \mid A \leftrightarrow Q \mid B$. □

LEMMA 10. *Suppose $(fr, th) \vdash M \leftrightarrow N$ and $m \notin fn(th)$. Then $m \in fn(M)$ if and only if $m \in fn(N)$.*

PROOF. By induction on the derivation of $(fr, th) \vdash M \leftrightarrow N$. □

LEMMA 11. *If $(fr \cup \{m\}, th) \vdash A \leftrightarrow B$ and $m \notin fr \cup fn(th)$ then $(fr, th) \vdash (\nu m)A \leftrightarrow (\nu m)B$.*

PROOF. If A and B are processes, the result follows at once using (Eq Res). Otherwise, $(fr, th) \vdash A \leftrightarrow B$ is derived via one of the following rules.

(Eq Abstraction) In this case, $A = (x)P'$ and $B = (x)Q'$ with $(fr \cup \{m\}, th) \vdash P' \leftrightarrow Q'$. We have both $(\nu m)(x)P' = (x)(\nu m)P'$ and $(\nu m)(x)Q' = (x)(\nu m)Q'$. By (Eq Res), $(fr \cup \{m\}, th) \vdash P' \leftrightarrow Q'$ and $m \notin fr \cup fn(th)$ imply $(fr, th) \vdash (\nu m)P' \leftrightarrow (\nu m)Q'$. By (Eq Abstraction), this implies $(fr, th) \vdash (x)(\nu m)P' \leftrightarrow (x)(\nu m)Q'$, that is, $(fr, th) \vdash (\nu m)A \leftrightarrow (\nu m)B$.

(Eq Concretion) In this case, $A = (\nu\vec{n})\langle M\rangle P'$ and $B = (\nu\vec{n})\langle N\rangle Q'$ with $(fr \cup \{m, \vec{n}\}, th) \vdash M \leftrightarrow N$, $(fr \cup \{m, \vec{n}\}, th) \vdash P' \leftrightarrow Q'$, and $\{\vec{n}\} \cap (fr \cup \{m\} \cup fn(th)) = \emptyset$. Now, either $m \in fn(M)$ or not.

- If $m \in \mathit{fn}(M)$, Lemma 10, $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$, and $m \notin \mathit{fn}(\mathit{th})$ imply that $m \in \mathit{fn}(N)$ too. Therefore we get that $(\nu m)A = (\nu m, \vec{n})\langle M\rangle P'$ and $(\nu m)B = (\nu m, \vec{n})\langle N\rangle Q'$. By (Eq Concretion), $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$, $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash P' \leftrightarrow Q'$, and $\{m, \vec{n}\} \cap (\mathit{fr} \cup \mathit{fn}(\mathit{th})) = \emptyset$ imply that $(\mathit{fr}, \mathit{th}) \vdash (\nu m, \vec{n})\langle M\rangle P' \leftrightarrow (\nu m, \vec{n})\langle N\rangle Q'$, that is, $(\mathit{fr}, \mathit{th}) \vdash (\nu m)A \leftrightarrow (\nu m)B$.
- If $m \notin \mathit{fn}(M)$, Lemma 10, $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$, and $m \notin \mathit{fn}(\mathit{th})$ imply that $m \notin \mathit{fn}(N)$ too. Therefore we get that $(\nu m)A = (\nu\vec{n})\langle M\rangle(\nu m)P'$ and $(\nu m)B = (\nu\vec{n})\langle N\rangle(\nu m)Q'$. By (Eq Res), $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash P' \leftrightarrow Q'$ and $m \notin \mathit{fr} \cup \{\vec{n}\} \cup \mathit{fn}(\mathit{th})$ imply $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash (\nu m)P' \leftrightarrow (\nu m)Q'$. By Lemma 6, $(\mathit{fr} \cup \{m, \vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$ and $m \notin \mathit{fn}(M) \cup \mathit{fn}(N)$ imply $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$. By (Eq Concretion), this, $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash (\nu m)P' \leftrightarrow (\nu m)Q'$, and $\{\vec{n}\} \cap (\mathit{fr} \cup \mathit{fn}(\mathit{th})) = \emptyset$ imply that $(\mathit{fr}, \mathit{th}) \vdash (\nu\vec{n})\langle M\rangle(\nu m)P' \leftrightarrow (\nu\vec{n})\langle N\rangle(\nu m)Q'$, that is, $(\mathit{fr}, \mathit{th}) \vdash (\nu m)A \leftrightarrow (\nu m)B$. □

LEMMA 12. *Suppose that* $(\mathit{fr}, \mathit{th}) \vdash P \leftrightarrow Q$ *and* $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$.

(1) *If* $P > P'$ *then there is a process* Q' *with* $Q > Q'$ *and* $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$.

(2) *If* $Q > Q'$ *then there is a process* P' *with* $P > P'$ *and* $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$.

PROOF. We prove part (1) by case analysis of the derivation of $P > P'$.

(Red Match) We have $P = [M\ \mathit{is}\ M]\ P' > P'$. Only (Eq Match) can derive $(\mathit{fr}, \mathit{th}) \vdash [M\ \mathit{is}\ M]\ P' \leftrightarrow Q$. Therefore $Q = [N_1\ \mathit{is}\ N_2]\ Q'$ with $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N_1$, $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N_2$, and $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$. By Lemma 4, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N_1$, and $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N_2$ imply that $N_1 = N_2$. Therefore, by (Red Match), $Q > Q'$.

(Red Let) We have $P = \mathit{let}\ (x, y) = (M, N)\ \mathit{in}\ R > R[M/x][N/y]$, together with $(\mathit{fr}, \mathit{th}) \vdash \mathit{let}\ (x, y) = (M, N)\ \mathit{in}\ R \leftrightarrow Q$. Only (Eq Let) can derive the latter. Therefore there are terms M' and N' and a process R' such that $Q = \mathit{let}\ (x, y) = (M', N')\ \mathit{in}\ R'$ with $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow M'$, $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N'$, and $(\mathit{fr}, \mathit{th}) \vdash R \leftrightarrow R'$. Let $Q' = R'[M'/x][N'/y]$. By two applications of Lemma 7(2) we obtain $(\mathit{fr}, \mathit{th}) \vdash R[M/x][N/y] \leftrightarrow Q'$ from $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow M'$, $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N'$, and $(\mathit{fr}, \mathit{th}) \vdash R \leftrightarrow R'$. Finally, by (Red Let), $Q > Q'$.

(Red Decrypt) We have $P = \mathit{case}\ \{M\}_N\ \mathit{of}\ \{x\}_N\ \mathit{in}\ R > R[M/x]$, together with $(\mathit{fr}, \mathit{th}) \vdash \mathit{case}\ \{M\}_N\ \mathit{of}\ \{x\}_N\ \mathit{in}\ R \leftrightarrow Q$. Only (Eq Decrypt) can derive the latter. Therefore there are terms L' and N' and a process R' such that $Q = \mathit{case}\ L'\ \mathit{of}\ \{x\}_{N'}\ \mathit{in}\ R'$ with $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N'$, $(\mathit{fr}, \mathit{th}) \vdash \{M\}_N \leftrightarrow L'$, and $(\mathit{fr}, \mathit{th}) \vdash R \leftrightarrow R'$. Since $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$ and $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N'$, we must have $(\mathit{fr}, \mathit{th}) \vdash \{M\}_N \leftrightarrow L'$ by (Eq

Encrypt) rather than by (Eq Theory). Therefore, $L' = \{M'\}_{N''}$ for some M' and N'', and $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow M'$ and $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N''$. Let $Q' = R'[M'/x]$. By Lemma 4, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N'$, and $(\mathit{fr}, \mathit{th}) \vdash N \leftrightarrow N''$ imply that $N' = N''$, so $Q > Q'$ by (Red Decrypt). Moreover, by Lemma 7(2), we obtain $(\mathit{fr}, \mathit{th}) \vdash R[M/x] \leftrightarrow Q'$ from $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow M'$ and $(\mathit{fr}, \mathit{th}) \vdash R \leftrightarrow R'$.

The other cases, which we omit, are similar. Part (2) follows by a symmetric argument. □

LEMMA 13. *Suppose that $(\mathit{fr}, \mathit{th}) \vdash P \leftrightarrow Q$ and $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$.*

(1) *If $P \xrightarrow{\alpha} A$ then there is an agent B with $Q \xrightarrow{\alpha} B$ and $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$. Moreover, if $\alpha = c$ or $\alpha = \overline{c}$ then $c \in \mathit{fr}$.*

(2) *If $Q \xrightarrow{\alpha} B$ then there is an agent A with $P \xrightarrow{\alpha} A$ and $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$. Moreover, if $\alpha = c$ or $\alpha = \overline{c}$ then $c \in \mathit{fr}$.*

PROOF. We prove part (1) by induction on the derivation of $P \xrightarrow{\alpha} A$.

(Comm In) We have $P = m(x).P'$, $\alpha = m$, and $A = (x)P'$. Only (Eq In) may derive $(\mathit{fr}, \mathit{th}) \vdash m(x).P' \leftrightarrow Q$. Therefore $Q = L(x).Q'$ with $(\mathit{fr}, \mathit{th}) \vdash m \leftrightarrow L$ and $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$. Since m is a name and $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, only (Eq Frame) may derive $(\mathit{fr}, \mathit{th}) \vdash m \leftrightarrow L$. Therefore $L = m$ and $m \in \mathit{fr}$. Let $B = (x)Q'$. By (Comm In), $Q = m(x).Q' \xrightarrow{m} B$. By (Eq Abstraction), $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$ implies $(\mathit{fr}, \mathit{th}) \vdash (x)P' \leftrightarrow (x)Q'$, that is, $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$.

(Comm Out) We have $P = \overline{m}\langle M\rangle.P'$, $\alpha = \overline{m}$, and $A = (\nu)\langle M\rangle P'$. Only (Eq Out) may derive $(\mathit{fr}, \mathit{th}) \vdash \overline{m}\langle M\rangle.P' \leftrightarrow Q$. Therefore $Q = \overline{L}\langle N\rangle.Q'$ with $(\mathit{fr}, \mathit{th}) \vdash m \leftrightarrow L$, $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N$, and $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$. Since m is a name and $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, only (Eq Frame) may derive $(\mathit{fr}, \mathit{th}) \vdash m \leftrightarrow L$. Therefore $L = m$ and $m \in \mathit{fr}$. Let $B = (\nu)\langle N\rangle Q'$. By (Comm Out), $Q = \overline{m}\langle N\rangle.Q' \xrightarrow{\overline{m}} B$. By (Eq Concretion), $(\mathit{fr}, \mathit{th}) \vdash M \leftrightarrow N$ and $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$ imply $(\mathit{fr}, \mathit{th}) \vdash (\nu)\langle M\rangle P' \leftrightarrow (\nu)\langle N\rangle Q'$, that is, $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$.

(Comm Inter 1) We have $P = P_1 \mid P_2$, $\alpha = \tau$, and $A = F@C$, where $P_1 \xrightarrow{m} F$ and $P_2 \xrightarrow{\overline{m}} C$. Only (Eq Par) may derive $(\mathit{fr}, \mathit{th}) \vdash P_1 \mid P_2 \leftrightarrow Q$. Therefore $Q = Q_1 \mid Q_2$ with $(\mathit{fr}, \mathit{th}) \vdash P_1 \leftrightarrow Q_1$ and $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$. By induction hypothesis, $(\mathit{fr}, \mathit{th}) \vdash P_1 \leftrightarrow Q_1$, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, and $P_1 \xrightarrow{m} F$ imply that there is an abstraction G such that $Q_1 \xrightarrow{m} G$ and $(\mathit{fr}, \mathit{th}) \vdash F \leftrightarrow G$. By induction hypothesis, $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, and $P_2 \xrightarrow{\overline{m}} C$ imply that there is a concretion D such that $Q_2 \xrightarrow{\overline{m}} D$ and $(\mathit{fr}, \mathit{th}) \vdash C \leftrightarrow D$. Let $B = G@D$. By (Comm Inter 1), $Q_1 \xrightarrow{m} G$ and $Q_2 \xrightarrow{\overline{m}} D$ imply that $Q_1 \mid Q_2 \xrightarrow{\tau} B$. By Lemma 8, $(\mathit{fr}, \mathit{th}) \vdash F \leftrightarrow G$ and $(\mathit{fr}, \mathit{th}) \vdash C \leftrightarrow D$ imply $(\mathit{fr}, \mathit{th}) \vdash F@C \leftrightarrow G@D$, that is, $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$.

(Comm Inter 2) Symmetric to the case of (Comm Inter 1).

(Comm Par 1) We have $P = P_1 \mid P_2$ and $A = A_1 \mid P_2$ where $P_1 \xrightarrow{\alpha} A_1$. Only (Eq Par) may derive $(\mathit{fr}, \mathit{th}) \vdash P_1 \mid P_2 \leftrightarrow Q$. Therefore $Q = Q_1 \mid Q_2$ with $(\mathit{fr}, \mathit{th}) \vdash P_1 \leftrightarrow Q_1$ and $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$. By induction hypothesis, $(\mathit{fr}, \mathit{th}) \vdash P_1 \leftrightarrow Q_1$, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, and $P_1 \xrightarrow{\alpha} A_1$ imply that there is an agent B_1 such that $Q_1 \xrightarrow{\alpha} B_1$ and $(\mathit{fr}, \mathit{th}) \vdash A_1 \leftrightarrow B_1$. Moreover, if $\alpha = c$ or $\alpha = \overline{c}$ then $c \in \mathit{fr}$. Let $B = B_1 \mid Q_2$. By (Comm Par 1), $Q_1 \xrightarrow{\alpha} B_1$ implies $Q_1 \mid Q_2 \xrightarrow{\alpha} B$. By Lemma 9, $(\mathit{fr}, \mathit{th}) \vdash A_1 \leftrightarrow B_1$ and $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$ imply $(\mathit{fr}, \mathit{th}) \vdash A_1 \mid P_2 \leftrightarrow B_1 \mid Q_2$, that is, $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$.

(Comm Par 2) Symmetric to the case of (Comm Inter 2).

(Comm Res) We have $P = (\nu m)P'$, $P' \xrightarrow{\alpha} A'$, $\alpha \notin \{m, \overline{m}\}$, and $A = (\nu m)A'$. Only (Eq Res) may derive $(\mathit{fr}, \mathit{th}) \vdash (\nu m)P' \leftrightarrow Q$. Therefore $Q = (\nu m)Q'$ with $m \notin \mathit{fr} \cup \mathit{fn}(\mathit{th})$ and $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash P' \leftrightarrow Q'$. We get $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash \mathit{ok}$ from $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$ and $m \notin \mathit{fr} \cup \mathit{fn}(\mathit{th})$ by Lemma 6. By induction hypothesis, $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash P' \leftrightarrow Q'$, $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash \mathit{ok}$, and $P' \xrightarrow{\alpha} A'$ imply that there is an agent B' such that $Q' \xrightarrow{\alpha} B'$ and $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash A' \leftrightarrow B'$. Moreover, if $\alpha = c$ or $\alpha = \overline{c}$ then $c \in \mathit{fr} \cup \{m\}$. Let $B = (\nu m)B'$. By (Comm Res), $Q' \xrightarrow{\alpha} B'$ and $\alpha \notin \{m, \overline{m}\}$ imply that $(\nu m)Q' \xrightarrow{\alpha} (\nu m)B'$, that is, $Q \xrightarrow{\alpha} B$. By Lemma 11, $(\mathit{fr} \cup \{m\}, \mathit{th}) \vdash A' \leftrightarrow B'$ and $m \notin \mathit{fr} \cup \mathit{fn}(\mathit{th})$ imply $(\mathit{fr}, \mathit{th}) \vdash (\nu m)A' \leftrightarrow (\nu m)B'$, that is, $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$. Finally, if $\alpha = c$ or $\alpha = \overline{c}$, $c \in \mathit{fr} \cup \{m\}$ and $\alpha \notin \{m, \overline{m}\}$ imply $c \in \mathit{fr}$.

(Comm Red) We have $P > P'$ and $P' \xrightarrow{\alpha} A$. By Lemma 12, $(\mathit{fr}, \mathit{th}) \vdash P \leftrightarrow Q$, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, and $P > P'$ imply that there is Q' with $Q > Q'$ and $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$. By induction hypothesis, $(\mathit{fr}, \mathit{th}) \vdash P' \leftrightarrow Q'$, $(\mathit{fr}, \mathit{th}) \vdash \mathit{ok}$, and $P' \xrightarrow{\alpha} A$ imply that there is an agent B such that $Q' \xrightarrow{\alpha} B$ and $(\mathit{fr}, \mathit{th}) \vdash A \leftrightarrow B$. Moreover, if $\alpha = c$ or $\alpha = \overline{c}$ then $c \in \mathit{fr}$. By (Comm Red), $Q > Q'$ and $Q' \xrightarrow{\alpha} B$ imply $Q \xrightarrow{\alpha} B$.

Part (2) follows by a symmetric argument. $\square$

Appendix A.5 Main Lemma

For any frame fr, theory th, and processes P and Q, let $\mathit{secrets}_1(\mathit{fr}, \mathit{th}, P)$ and $\mathit{secrets}_2(\mathit{fr}, \mathit{th}, Q)$ be as follows:

$$\begin{aligned} \mathit{secrets}_1(\mathit{fr}, \mathit{th}, P) &\triangleq (\mathit{fn}(P) \cup \mathit{fn}(\pi_1(\mathit{th}))) - \mathit{fr} \\ \mathit{secrets}_2(\mathit{fr}, \mathit{th}, Q) &\triangleq (\mathit{fn}(Q) \cup \mathit{fn}(\pi_2(\mathit{th}))) - \mathit{fr} \end{aligned}$$

LEMMA 14. *Suppose $\mathcal{S}$ is the relation such that $P \mathrel{\mathcal{S}} Q$ if and only if there is a frame fr, a theory th, and processes P_1, P_2, Q_1, Q_2, such that*

$$\begin{array}{rcl} P & = & (\nu\vec{p})(P_1 \mid P_2) \\ Q & = & (\nu\vec{q})(Q_1 \mid Q_2) \end{array}$$

and $(fr, th) \vdash ok$, $(fr, th) \vdash P_1 \sim_f Q_1$, and $(fr, th) \vdash P_2 \leftrightarrow Q_2$, where $\{\vec{p}\} = secrets_1(fr, th, P_1)$ and $\{\vec{q}\} = secrets_2(fr, th, Q_1)$. Then, if $P \mathrel{\mathcal{S}} Q$:

(1) *For any barb β, if there is A with $P \xrightarrow{\beta} A$ then there is B with $Q \xrightarrow{\beta} B$.*
(2) *If $P \xrightarrow{\tau} P'$ then there exists $\vec{r}$, P'', Q', and Q'' such that $Q \xrightarrow{\tau} Q'$, $P' \equiv (\nu\vec{r})P''$, $Q' \equiv (\nu\vec{r})Q''$, and $P'' \mathrel{\mathcal{S}} Q''$.*

PROOF. Suppose $P \mathrel{\mathcal{S}} Q$, that is, $P = (\nu\vec{p})(P_1 \mid P_2)$ and $Q = (\nu\vec{q})(Q_1 \mid Q_2)$, where $(fr, th) \vdash ok$, $(fr, th) \vdash P_1 \sim_f Q_1$, $(fr, th) \vdash P_2 \leftrightarrow Q_2$, and:

$$\begin{array}{rcl} \{\vec{p}\} & = & secrets_1(fr, th, P_1) = (fn(P_1) \cup fn(\pi_1(th))) - fr \\ \{\vec{q}\} & = & secrets_2(fr, th, Q_1) = (fn(Q_1) \cup fn(\pi_2(th))) - fr \end{array}$$

By Lemma 3, $fn(P_2) \subseteq fr \cup fn(\pi_1(th))$. Since $fn(P) = (fn(P_1) \cup fn(P_2)) - secrets_1(fr, th, P_1)$, it follows that $fn(P) \subseteq fr$. Similarly, we have $fn(Q) \subseteq fr$.

For part (1), consider any commitment $(\nu\vec{p})(P_1 \mid P_2) \xrightarrow{\beta} A$ for some barb $\beta \in \{c, \overline{c}\}$. It must be that $c \in fn((\nu\vec{p})(P_1 \mid P_2))$ and therefore that $c \in fr$. Since (Comm Red) does not apply to a restriction or a composition, the derivation of this commitment must end with a use of (Comm Res) for each name in $\vec{p}$, followed by one of the following rules.

(Comm Par 1) In this case, there is A' such that $P_1 \xrightarrow{\beta} A'$ and $A = (\nu\vec{p})(A' \mid P_2)$. Since $(fr, th) \vdash P_1 \sim_f Q_1$, and $\sim_f$ is a framed bisimulation, $P_1 \xrightarrow{\beta} A'$ and $c \in fr$, together with the fact that we may assume that the names bound in any outermost restrictions in A' are disjoint from $fn(P_1) \cup fn(\pi_1(th)) \cup fr$, imply that there is B' such that $Q_1 \xrightarrow{\beta} B'$. Since $c \in fr$ it follows that $c \notin \{\vec{q}\}$. So using (Comm Res) and (Comm Par 1) we have $Q \xrightarrow{\beta} (\nu\vec{q})(B' \mid Q_2)$.

(Comm Par 2) In this case, there is A' such that $P_2 \xrightarrow{\beta} A'$ and $A = (\nu\vec{p})(P_1 \mid A')$. By Lemma 13(1), $(fr, th) \vdash P_2 \leftrightarrow Q_2$ and $P_2 \xrightarrow{\beta} A'$ imply that there is B' such that $Q_2 \xrightarrow{\beta} B'$. Since $c \in fr$ it follows that $c \notin \{\vec{q}\}$. So using (Comm Res) and (Comm Par 1) we have $Q \xrightarrow{\beta} (\nu\vec{q})(Q_1 \mid B')$.

For part (2), consider any commitment $(\nu\vec{p})(P_1 \mid P_2) \xrightarrow{\tau} P'$. The derivation of such a commitment must end with a use of (Comm Res) for each name in $\vec{p}$, and one of the following rules.

(Comm Inter 1) In this case, there is a name c, an abstraction $F = (x)P_1'$, and a concretion $C = (\nu\vec{n})\langle M\rangle P_2'$, such that $P_1 \xrightarrow{c} F$, $P_2 \xrightarrow{\overline{c}} C$, and $P' = (\nu\vec{p})(F@C) = (\nu\vec{p})(\nu\vec{n})(P_1'[M/x] \mid P_2')$. We may assume that the set of bound names $\{\vec{n}\}$ is disjoint from $\mathit{fr} \cup \{\vec{p}\} \cup \{\vec{q}\} \cup \mathit{fn}(\mathit{th})$.

By Lemma 13(1), $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$ and $P_2 \xrightarrow{\overline{c}} C$ imply that $c \in \mathit{fr}$, and there is a concretion D such that $Q_2 \xrightarrow{\overline{c}} D$ with $(\mathit{fr}, \mathit{th}) \vdash C \leftrightarrow D$. The derivation of the latter must end with (Eq Concretion). Therefore there is a term N and a process Q_2' such that $D = (\nu\vec{n})\langle N\rangle Q_2'$, and $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$ and $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$.

Since $(\mathit{fr}, \mathit{th}) \vdash P_1 \sim_f Q_1$, and $\sim_f$ is a framed bisimulation, $P_1 \xrightarrow{c} F$ and $c \in \mathit{fr}$ imply that there is $G = (x)Q_1'$ with $Q_1 \xrightarrow{c} G$, and, since $\{\vec{n}\}$ is disjoint from $\mathit{fr} \cup \{\vec{p}\} \cup \{\vec{q}\} \cup \mathit{fn}(\mathit{th})$ we have $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash \mathit{ok}$ and $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash F \,\overline{\sim}_f\, G$. The latter together with $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash M \leftrightarrow N$ imply $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash P_1'[M/x] \sim_f Q_1'[N/x]$.

Let $\{\vec{p}_{\text{after}}\} = \mathit{secrets}_1(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}, P_1'[M/x])$ and $\{\vec{q}_{\text{after}}\} = \mathit{secrets}_2(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}, Q_1'[N/x])$. Since $\mathit{fn}(P_1') \subseteq \mathit{fn}(P_1)$ and $\mathit{fn}(M) \subseteq \mathit{fr} \cup \{\vec{n}\} \cup \mathit{fn}(\pi_1(\mathit{th}))$ by Lemma 3, it follows that:

$$
\begin{array}{rcl}
\{\vec{p}_{\text{after}}\} & = & (\mathit{fn}(P_1'[M/x]) \cup \mathit{fn}(\pi_1(\mathit{th}))) - (\mathit{fr} \cup \{\vec{n}\}) \\
& \subseteq & (\mathit{fn}(P_1) \cup \mathit{fr} \cup \{\vec{n}\} \cup \mathit{fn}(\pi_1(\mathit{th}))) - (\mathit{fr} \cup \{\vec{n}\}) \\
& = & (\mathit{fn}(P_1) \cup \mathit{fn}(\pi_1(\mathit{th}))) - (\mathit{fr} \cup \{\vec{n}\}) \\
& \subseteq & \{\vec{p}\}
\end{array}
$$

By symmetric reasoning, $\{\vec{q}_{\text{after}}\} \subseteq \{\vec{q}\}$.

If we make the following definitions,

$$
\begin{array}{rcl}
P'' & = & (\nu\vec{p}_{\text{after}})(P_1'[M/x] \mid P_2') \\
Q'' & = & (\nu\vec{q}_{\text{after}})(Q_1'[N/x] \mid Q_2')
\end{array}
$$

we have $P''\, \mathcal{S}\, Q''$, since $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash \mathit{ok}$, $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash P_1'[M/x] \sim_f Q_1'[N/x]$, and $(\mathit{fr} \cup \{\vec{n}\}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$.

Let $\{\vec{p}_{\text{lost}}\} = \mathit{fn}(P_1) - (\mathit{fn}(\pi_1(\mathit{th})) \cup \mathit{fn}(P_1'[M/x]) \cup \mathit{fr})$ and $\{\vec{q}_{\text{lost}}\} = (\mathit{fn}(Q_1) - (\mathit{fn}(\pi_2(\mathit{th})) \cup \mathit{fn}(Q_1'[N/x]) \cup \mathit{fr})$. We have:

$$
\begin{array}{rcl}
\{\vec{p}_{\text{lost}}\} & = & \{\vec{p}\} - \{\vec{p}_{\text{after}}\} \\
\{\vec{q}_{\text{lost}}\} & = & \{\vec{q}\} - \{\vec{q}_{\text{after}}\}
\end{array}
$$

and that $\{\{\vec{p}_{\text{after}}\}, \{\vec{p}_{\text{lost}}\}\}$ partitions $\{\vec{p}\}$ and $\{\{\vec{q}_{\text{after}}\}, \{\vec{q}_{\text{lost}}\}\}$ partitions $\{\vec{q}\}$. Moreover, $\mathit{fn}(P'') \cap \{\vec{p}_{\text{lost}}\} = \emptyset$ and $\mathit{fn}(Q'') \cap \{\vec{q}_{\text{lost}}\} = \emptyset$.

Let $Q' = (\nu\vec{q})(G@D)$. Using (Comm Res) and (Comm Inter 1), we have $Q \xrightarrow{\tau} Q'$. We have

$$
P' \quad = \quad (\nu\vec{p})(\nu\vec{n})(P_1'[M/x] \mid P_2')
$$

$$\begin{array}{rcl} & \equiv & (\nu\vec{p}_{\mathrm{lost}})(\nu\vec{p}_{\mathrm{after}})(\nu\vec{n})(P_1'[M/x] \mid P_2') \\ & \equiv & (\nu\vec{n})P'' \\ Q' & = & (\nu\vec{q})(\nu\vec{n})(Q_1'[N/x] \mid Q_2') \\ & \equiv & (\nu\vec{q}_{\mathrm{lost}})(\nu\vec{q}_{\mathrm{after}})(\nu\vec{n})(Q_1'[N/x] \mid Q_2') \\ & \equiv & (\nu\vec{n})Q'' \end{array}$$

which completes the case.

(Comm Inter 2) In this case, there is a name c, a concretion $C = (\nu\vec{m})\langle M\rangle P_1'$, and an abstraction $F = (x)P_2'$, such that $P_1 \xrightarrow{\bar{c}} C$, $P_2 \xrightarrow{c} F$, and $P' = (\nu\vec{p})(C@F) = (\nu\vec{p})(\nu\vec{m})(P_1' \mid P_2'[M/x])$.

By Lemma 13(1), $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$ and $P_2 \xrightarrow{c} F$ imply that $c \in \mathit{fr}$, and there is an abstraction G such that $Q_2 \xrightarrow{c} G$ with $(\mathit{fr}, \mathit{th}) \vdash F \leftrightarrow G$. The derivation of the latter must end with (Eq Abstraction), so there is Q_2' such that $G = (x)Q_2'$ and $(\mathit{fr}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$.

Since the names $\vec{m}$ are bound, we may assume that the set $\{\vec{m}\}$ is disjoint from $\mathit{fr} \cup \{\vec{p}\}$. Since $(\mathit{fr}, \mathit{th}) \vdash P_1 \sim_f Q_1$, and $\sim_f$ is a framed bisimulation, $P_1 \xrightarrow{\bar{c}} (\nu\vec{m})\langle M\rangle P_1'$, $c \in \mathit{fr}$, and $\{\vec{m}\}$ disjoint from $\mathit{fr} \cup \{\vec{p}\}$ imply that there is $D = (\nu\vec{n})\langle N\rangle Q_1'$ such that $Q_1 \xrightarrow{\bar{c}} D$, set $\{\vec{n}\}$ is disjoint from $\mathit{fr} \cup \{\vec{q}\}$, and $(\mathit{fr}, \mathit{th}) \vdash \langle M\rangle P_1' \overline{\sim}_f \langle N\rangle Q_1'$. The latter implies that there is a frame fr' and a theory th' such that $(\mathit{fr}, \mathit{th}) \leq (\mathit{fr}', \mathit{th}')$, $(\mathit{fr}', \mathit{th}') \vdash \mathit{ok}$, $(\mathit{fr}', \mathit{th}') \vdash M \leftrightarrow N$, and $(\mathit{fr}', \mathit{th}') \vdash P_1' \sim_f Q_1'$. Given $(\mathit{fr}, \mathit{th}) \leq (\mathit{fr}', \mathit{th}')$, $(\mathit{fr}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$ implies $(\mathit{fr}', \mathit{th}') \vdash P_2' \leftrightarrow Q_2'$. Lemma 7 yields $(\mathit{fr}', \mathit{th}') \vdash P_2'[M/x] \leftrightarrow Q_2'[N/x]$. Let $Q' = (\nu\vec{q})(D@G) = (\nu\vec{q})(\nu\vec{n})(Q_1' \mid Q_2'[N/x])$. By (Comm Inter 2), $Q \xrightarrow{\tau} Q'$.

If we make the definitions,

$$\begin{array}{rcl} \{\vec{p}_{\mathrm{after}}\} & \triangleq & \mathit{secrets}_1(\mathit{fr}', \mathit{th}', P_1') \\ \{\vec{q}_{\mathrm{after}}\} & \triangleq & \mathit{secrets}_2(\mathit{fr}', \mathit{th}', Q_1') \\ P'' & \triangleq & (\nu\vec{p}_{\mathrm{after}})(P_1' \mid P_2'[M/x]) \\ Q'' & \triangleq & (\nu\vec{q}_{\mathrm{after}})(Q_1' \mid Q_2'[N/x]) \end{array}$$

we have $P'' \mathrel{\mathcal{S}} Q''$ from $(\mathit{fr}', \mathit{th}') \vdash \mathit{ok}$, $(\mathit{fr}', \mathit{th}') \vdash P_1' \sim_f Q_1'$, and $(\mathit{fr}', \mathit{th}') \vdash P_2'[M/x] \leftrightarrow Q_2'[N/x]$.

By Proposition 1, there must be a set $\{\vec{r}\}$, disjoint from fr, such that $\mathit{fr}' = \mathit{fr} \cup \{\vec{r}\}$. Now, if the following equations are instances of structural equivalence,

$$\begin{array}{rcll} P' = (\nu\vec{p})(\nu\vec{m})(P_1' \mid P_2'[M/x]) & \equiv & (\nu\vec{r})(\nu\vec{p}_{\mathrm{after}})(P_1' \mid P_2'[M/x]) & (1) \\ Q' = (\nu\vec{q})(\nu\vec{n})(Q_1' \mid Q_2'[N/x]) & \equiv & (\nu\vec{r})(\nu\vec{q}_{\mathrm{after}})(Q_1' \mid Q_2'[N/x]) & (2) \end{array}$$

we obtain $P' \equiv (\nu\vec{r})P''$ and $Q' \equiv (\nu\vec{r})Q''$, and this case is complete.

We now prove equation (1). We begin by defining a series of sets with which each of $\{\vec{p}\}$, $\{\vec{m}\}$, $\{\vec{r}\}$, and $\{\vec{p}_{\text{after}}\}$ may be partitioned.

- Let $\{\vec{p}_{\text{pub}}\}$ be the names in $\{\vec{p}\}$ made public in the new frame; we divide the names in $\{\vec{p}\}$ that remain secret into: $\{\vec{p}_{\text{prv}}\}$, the names that remain relevant to P_1' and $\pi_1(th')$, and $\{\vec{p}_{\text{lost}}\}$, the names that have been dropped.

$$\begin{array}{rcl}\{\vec{p}_{\text{pub}}\} & \triangleq & \{\vec{p}\} \cap \{\vec{r}\}\\ \{\vec{p}_{\text{prv}}\} & \triangleq & (\{\vec{p}\} - \{\vec{r}\}) \cap (\mathit{fn}(P_1') \cup \mathit{fn}(\pi_1(th')))\\ \{\vec{p}_{\text{lost}}\} & \triangleq & (\{\vec{p}\} - \{\vec{r}\}) - (\mathit{fn}(P_1') \cup \mathit{fn}(\pi_1(th')))\end{array}$$

(Partition $\vec{p}$): The set $\{\{\vec{p}_{\text{pub}}\}, \{\vec{p}_{\text{prv}}\}, \{\vec{p}_{\text{lost}}\}\}$ partitions $\{\vec{p}\}$.
This holds by definition.

(Disjoint $\vec{p}_{\text{lost}}$): $\{\vec{p}_{\text{lost}}\}$ and $\mathit{fn}(P_1' \mid P_2'[M/x])$ are disjoint.
Since $(\mathit{fr} \cup \{\vec{r}\}, th') \vdash P_2'[M/x] \leftrightarrow Q_2'[N/x]$, Lemma 3 implies $\mathit{fn}(P_2'[M/x]) \subseteq \mathit{fr} \cup \{\vec{r}\} \cup \mathit{fn}(\pi_1(th'))$. We have

$$\begin{array}{rcl}\mathit{fn}(P_1' \mid P_2'[M/x]) & \subseteq & \mathit{fr} \cup \mathit{fn}(P_1') \cup \{\vec{r}\} \cup \mathit{fn}(\pi_1(th'))\\ \{\vec{p}_{\text{lost}}\} & = & \{\vec{p}\} - (\mathit{fn}(P_1') \cup \{\vec{r}\} \cup \mathit{fn}(\pi_1(th')))\end{array}$$

Since fr is disjoint from $\{\vec{p}\}$, the claim follows.

- Let $\{\vec{m}_{\text{pub}}\}$ be the names in $\{\vec{m}\}$ revealed in the new frame, and let $\{\vec{m}_{\text{prv}}\}$ be the names in $\{\vec{m}\}$ that remain secret.

$$\begin{array}{rcl}\{\vec{m}_{\text{pub}}\} & \triangleq & \{\vec{m}\} \cap \{\vec{r}\}\\ \{\vec{m}_{\text{prv}}\} & \triangleq & \{\vec{m}\} - \{\vec{r}\}\end{array}$$

(Partition $\vec{m}$): The set $\{\{\vec{m}_{\text{pub}}\}, \{\vec{m}_{\text{prv}}\}\}$ partitions $\{\vec{m}\}$.
This holds by definition.

(Subset $\vec{m}$): $\{\vec{m}\} \subseteq \{\vec{r}\} \cup \mathit{fn}(\pi_1(th'))$.
Since $(\mathit{fr} \cup \{\vec{r}\}, th') \vdash M \leftrightarrow N$, Lemma 3 implies $\mathit{fn}(M) \subseteq \mathit{fr} \cup \{\vec{r}\} \cup \mathit{fn}(\pi_1(th'))$. The claim follows since we know $\{\vec{m}\} \subseteq \mathit{fn}(M)$, and that fr and $\{\vec{m}\}$ are disjoint.

- Let $\{\vec{r}_{\text{new}}\}$ be the names in $\{\vec{r}\}$ that do not appear in $\{\vec{p}\}$ or $\{\vec{m}\}$.

$$\{\vec{r}_{\text{new}}\} \triangleq \{\vec{r}\} - (\{\vec{p}\} \cup \{\vec{m}\})$$

(Partition $\vec{r}$): The set $\{\{\vec{p}_{\text{pub}}\}, \{\vec{m}_{\text{pub}}\}, \{\vec{r}_{\text{new}}\}\}$ partitions $\{\vec{r}\}$.
This holds since $\{\vec{p}\}$ and $\{\vec{m}\}$ are disjoint.

(Disjoint $\vec{r}_{\text{new}}$): $\{\vec{r}_{\text{new}}\}$ and $\mathit{fn}(P_1' \mid P_2'[M/x])$ are disjoint.
Since fr and $\{\vec{r}\}$ are disjoint, we have:

$$\{\vec{r}_{\text{new}}\} = \{\vec{r}\} - (\mathit{fr} \cup \{\vec{p}\} \cup \{\vec{m}\})$$

Since $fn(P) \subseteq fr$, as established near the start of the proof of this lemma, we have $fn(P_1 \mid P_2) \subseteq fr \cup \{\vec{p}\}$. Since $P_1 \mid P_2 \xrightarrow{\tau} (\nu\vec{m})(P_1' \mid P_2'[M/x])$, it follows that

$$\begin{array}{rcl} fn(P_1' \mid P_2'[M/x]) & \subseteq & fn(P_1 \mid P_2) \cup \{\vec{m}\} \\ & \subseteq & fr \cup \{\vec{p}\} \cup \{\vec{m}\} \end{array}$$

Therefore the claim holds.

- Let $\{\vec{s}_{\text{new}}\}$ be the fresh names introduced by th'.

$$\{\vec{s}_{\text{new}}\} \triangleq fn(\pi_1(th')) - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr')$$

(Partition $\vec{p}_{\text{after}}$): The set $\{\{\vec{p}_{\text{prv}}\}, \{\vec{m}_{\text{prv}}\}, \{\vec{s}_{\text{new}}\}\}$ partitions $\{\vec{p}_{\text{after}}\}$.
Let $X = fn(P_1') \cup fn(\pi_1(th'))$. By definition, and because $\{\vec{p}\} \cap fr' = \{\vec{p}\} \cap \{\vec{r}\}$ since $\{\vec{p}\} \cap fr = \emptyset$, we have:

$$\begin{array}{rcl} \{\vec{p}_{\text{after}}\} & = & X - fr' \\ \{\vec{p}_{\text{prv}}\} & = & X \cap (\{\vec{p}\} - fr') \end{array} \tag{3}$$

Property (Subset $\vec{m}$), that is, $\{\vec{m}\} \subseteq \{\vec{r}\} \cup fn(\pi_1(th'))$, implies that $\{\vec{m}_{\text{prv}}\} = \{\vec{m}\} - \{\vec{r}\} \subseteq X$. Therefore, since $\{\vec{m}\} \cap fr = \emptyset$, we have:

$$\{\vec{m}_{\text{prv}}\} = X \cap (\{\vec{m}\} - fr')$$

By combining the last two equations we obtain:

$$\{\vec{p}_{\text{prv}}\} \cup \{\vec{m}_{\text{prv}}\} = X \cap ((\{\vec{p}\} \cup \{\vec{m}\}) - fr') \tag{4}$$

Since $fn(P_1') \subseteq fr \cup \{\vec{p}\} \cup \{\vec{m}\}$, we have

$$\{\vec{s}_{\text{new}}\} = X - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr') \tag{5}$$

At this point, we notice that the following is tautological:

$$\begin{array}{rcl} X - fr' & = & (X \cap ((\{\vec{p}\} \cup \{\vec{m}\}) - fr')) \cup \\ & & (X - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr')) \end{array}$$

- $lhs \subseteq rhs$
  Suppose $n \in X$ and $n \notin fr'$. Either $n \in \{\vec{p}\} \cup \{\vec{m}\}$ or not. If so, $n \in X \cap ((\{\vec{p}\} \cup \{\vec{m}\}) - fr')$. If not, $n \in X - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr')$. In either case, $n \in rhs$.
- $rhs \subseteq lhs$
  Either $n \in X \cap ((\{\vec{p}\} \cup \{\vec{m}\}) - fr')$ or $n \in X - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr')$. In either case $n \in X$ and $n \notin fr'$. Therefore $n \in lhs$.

Given this tautology, equations (3), (4), and (5) imply that $\{\vec{p}_{\text{after}}\}$ is the union of the set $\{\{\vec{p}_{\text{prv}}\}, \{\vec{m}_{\text{prv}}\}, \{\vec{s}_{\text{new}}\}\}$. Since the members of this set are clearly pairwise disjoint, it partitions $\{\vec{p}_{\text{after}}\}$.

(Disjoint $\vec{s}_{\text{new}}$): $\{\vec{s}_{\text{new}}\}$ and $fn(P_1' \mid P_2'[M/x])$ are disjoint.
We know that $fn(P_1' \mid P_2'[M/x]) \subseteq fr \cup \{\vec{p}\} \cup \{\vec{m}\}$ and $\{\vec{s}_{\text{new}}\} = X - (\{\vec{p}\} \cup \{\vec{m}\} \cup fr')$.

We can now calculate equation (1):

$$
\begin{aligned}
&(\nu\vec{p})(\nu\vec{m})(P_1' \mid P_2'[M/x]) \\
&\equiv (\nu\vec{p}_{\text{pub}})(\nu\vec{p}_{\text{prv}})(\nu\vec{p}_{\text{lost}})(\nu\vec{m}_{\text{pub}})(\nu\vec{m}_{\text{prv}})(P_1' \mid P_2'[M/x]) \qquad (6)\\
&\equiv (\nu\vec{p}_{\text{pub}})(\nu\vec{m}_{\text{pub}})(\nu\vec{r}_{\text{new}})(\nu\vec{p}_{\text{prv}})(\nu\vec{m}_{\text{prv}})(\nu\vec{s}_{\text{new}})(P_1' \mid P_2'[M/x]) \qquad (7)\\
&\equiv (\nu\vec{r})(\nu\vec{p}_{\text{after}})(P_1' \mid P_2'[M/x]) \qquad (8)
\end{aligned}
$$

Step (6) follows from (Partition $\vec{p}$) and (Partition $\vec{m}$). In step (7), we drop $\{\vec{p}_{\text{lost}}\}$ using (Disjoint $\vec{p}_{\text{lost}}$), introduce $\{\vec{r}_{\text{new}}\}$ and $\{\vec{s}_{\text{new}}\}$ using (Disjoint $\vec{r}_{\text{new}}$) and (Disjoint $\vec{s}_{\text{new}}$) respectively, and permutation of restrictions. Finally, step (8) follows from (Partition $\vec{r}$) and (Partition $\vec{p}_{\text{after}}$).

Equation (2) can be proved by exactly symmetric considerations. This case is now complete.

(Comm Par 1) In this case, there is P_1' such that $P_1 \xrightarrow{\tau} P_1'$ and $P' = (\nu\vec{p})(P_1' \mid P_2)$. Since $(fr, th) \vdash P_1 \sim_f Q_1$, and $\sim_f$ is a framed bisimulation, $P_1 \xrightarrow{\tau} P_1'$ implies that there is Q_1' with $Q_1 \xrightarrow{\tau} Q_1'$, and $(fr, th) \vdash P_1' \sim_f Q_1'$. Let $Q' = (\nu\vec{p})(Q_1' \mid Q_2)$. By (Comm Res) and (Comm Par 1), $Q \xrightarrow{\tau} Q'$.

If we make the definitions,

$$
\begin{aligned}
\{\vec{p}_{\text{after}}\} &\triangleq secrets_1(fr, th, P_1')\\
\{\vec{q}_{\text{after}}\} &\triangleq secrets_2(fr, th, Q_1')\\
P'' &\triangleq (\nu\vec{p}_{\text{after}})(P_1' \mid P_2)\\
Q'' &\triangleq (\nu\vec{q}_{\text{after}})(Q_1' \mid Q_2)
\end{aligned}
$$

we have $P'' \mathcal{S} Q''$ from $(fr, th) \vdash P_1' \sim_f Q_1'$.

Note that:

$$
\begin{aligned}
\{\vec{p}\} &= (fn(P_1) \cup fn(\pi_1(th))) - fr\\
\{\vec{p}_{\text{after}}\} &= (fn(P_1') \cup fn(\pi_1(th))) - fr\\
fn(P_1') &\subseteq fn(P_1)\\
fn(P_2) &\subseteq fr \cup fn(\pi_1(th))
\end{aligned}
$$

Therefore $\{\vec{p}_{\text{after}}\} \subseteq \{\vec{p}\}$. Let $\{\vec{p}_{\text{lost}}\} = \{\vec{p}\} - \{\vec{p}_{\text{after}}\}$, so that the set $\{\{\vec{p}_{\text{after}}\}, \{\vec{p}_{\text{lost}}\}\}$ partitions $\{\vec{p}\}$, and we have:

$$\begin{aligned}
\{\vec{p}_{\text{lost}}\} &= ((\mathit{fn}(P_1) \cup \mathit{fn}(\pi_1(\mathit{th}))) - \mathit{fr}) - ((\mathit{fn}(P_1') \cup \mathit{fn}(\pi_1(\mathit{th}))) - \mathit{fr}) \\
&= ((\mathit{fn}(P_1) \cup \mathit{fn}(\pi_1(\mathit{th}))) - \mathit{fr}) - (\mathit{fn}(P_1') \cup \mathit{fn}(\pi_1(\mathit{th}))) \\
&= \mathit{fn}(P_1) - (\mathit{fr} \cup \mathit{fn}(P_1') \cup \mathit{fn}(\pi_1(\mathit{th})))
\end{aligned}$$

Hence, by Lemma 3, we have that $\{\vec{p}_{\text{lost}}\}$ is disjoint from $\mathit{fn}(P_1' \mid P_2)$, and we may calculate:

$$\begin{aligned}
P' &\equiv (\nu\vec{p}_{\text{after}})(\nu\vec{p}_{\text{lost}})(P_1' \mid P_2) \\
&\equiv (\nu\vec{p}_{\text{after}})(P_1' \mid P_2) \\
&= P''
\end{aligned}$$

To complete this case we need only prove $Q' \equiv Q''$, but this follows by appeal to symmetry.

(Comm Par 2) In this case, there is P_2' such that $P_2 \xrightarrow{\tau} P_2'$ and $P' = P_1 \mid P_2'$. By Lemma 13(1), $(\mathit{fr}, \mathit{th}) \vdash P_2 \leftrightarrow Q_2$ and $P_2 \xrightarrow{\tau} P_2'$ imply that there is a process Q_2' with $Q_2 \xrightarrow{\tau} Q_2'$ and $(\mathit{fr}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$. Let $Q' = (\nu\vec{p})(Q_1 \mid Q_2')$. By (Comm Res) and (Comm Par 2), $Q \xrightarrow{\tau} Q_1'$. Since $\{\vec{p}\} = \mathit{secrets}_1(\mathit{fr}, \mathit{th}, P_1)$ and $(\mathit{fr}, \mathit{th}) \vdash P_2' \leftrightarrow Q_2'$, it follows by definition of $\mathcal{S}$ that $(\nu\vec{p})(P_1 \mid P_2')\ \mathcal{S}\ Q'$. This completes the proof of this final case of the argument for part (2) of the lemma.□

COROLLARY 2. *Let $\mathcal{S}$ be as in Lemma 14. Then $\mathcal{S} \subseteq \dot{\sim}$.*

PROOF. Since the relation $\mathcal{S}$ defined in Lemma 14 is symmetric, $\mathcal{S}$ is a barbed bisimulation up to restriction and $\dot{\sim}$, and therefore $\mathcal{S} \subseteq \dot{\sim}$ [3]. □

Appendix A.6 Soundness Theorems

The soundness theorems of this section say that framed bisimulation is sound with respect to barbed congruence and testing equivalence. Their proofs are fairly direct applications of Corollary 2.

THEOREM 2. *Consider any closed processes P and Q. Suppose that $(\mathit{fr}, \emptyset) \vdash P \sim_f Q$ for all finite fr with $\mathit{fn}(P) \cup \mathit{fn}(Q) \subseteq \mathit{fr}$. Then $P \sim Q$.*

PROOF. Consider any closed process R. To prove the theorem, we need to show that $P \mid R \dot{\sim} Q \mid R$. Let $\mathit{fr} = \mathit{fn}(P) \cup \mathit{fn}(Q) \cup \mathit{fn}(R)$. We have $(\mathit{fr}, \emptyset) \vdash \mathit{ok}$, $(\mathit{fr}, \emptyset) \vdash P \sim_f Q$, $(\mathit{fr}, \emptyset) \vdash R \leftrightarrow R$ by Lemma 2, and $\mathit{secrets}_1(\mathit{fr}, \emptyset, P) = \mathit{secrets}_2(\mathit{fr}, \emptyset, Q) = \emptyset$. Therefore $(P \mid R)\ \mathcal{S}\ (Q \mid R)$, where $\mathcal{S}$ is the relation defined in the statement of Lemma 14, so $P \mid R \dot{\sim} Q \mid R$ by Corollary 2. □

PROOF OF THEOREM 1 *Consider any closed processes P and Q, and any name $n \notin fn(P) \cup fn(Q)$. Suppose that $(fn(P) \cup fn(Q) \cup \{n\}, \emptyset) \vdash P \sim_f Q$. Then $P \simeq Q$.*

PROOF. According to Corollary 1, we can establish $P \simeq Q$ by proving that for every name n, every barb $\beta \in \{m, \overline{m}\}$, and every R such that $fn(R) \subseteq fn(P) \cup fn(Q) \cup \{m\}$,

$$P \mid R \xrightarrow{\tau}{}^{*} P' \xrightarrow{\beta} A$$

for some P' and A if and only if

$$Q \mid R \xrightarrow{\tau}{}^{*} Q' \xrightarrow{\beta} B$$

for some Q' and B. By a standard renaming, we can take $m = n$ if $m \notin fn(P) \cup fn(Q)$. Let $fr = fn(P) \cup fn(Q) \cup \{n\}$. We have $(fr, \emptyset) \vdash ok$, $(fr, \emptyset) \vdash P \sim_f Q$, $(fr, \emptyset) \vdash R \leftrightarrow R$ by Lemma 2, and $secrets_1(fr, \emptyset, P) = secrets_2(fr, \emptyset, Q) = \emptyset$. Therefore $(P \mid R)\ \mathcal{S}\ (Q \mid R)$, where $\mathcal{S}$ is the relation defined in the statement of Lemma 14, so $P \mid R \stackrel{\bullet}{\sim} Q \mid R$ by Corollary 2. The desired result follows from Proposition 2. □

Unlike our theorem for testing equivalence, our theorem for barbed congruence considers an arbitrary finite frame fr with $fn(P) \cup fn(Q) \subseteq fr$. It may be possible to restrict attention to a fixed fr. Moreover, it may be possible to strengthen both soundness theorems by restricting attention to the frame $fn(P) \cup fn(Q)$ (without any fresh name n). We have not verified these minor improvements.

Appendix A.7 Incompleteness

As noted in Section 3.4, framed bisimilarity is not a necessary condition for testing equivalence, for rather simple reasons. We conjecture that, in fact, framed bisimilarity is not even a necessary condition for barbed congruence. Thus, we do not have a converse to Theorem 2.

Our conjecture is based on the following example. We let the processes P and Q be defined by:

$$\begin{array}{rcl} P & \triangleq & (\nu K)(\nu K_{01})\overline{c}\langle\{K_{01}\}_K\rangle.c(x).P' \\ Q & \triangleq & (\nu K)(\nu K_0)(\nu K_1)\overline{c}\langle\{K_0, K_1\}_K\rangle.c(x).Q' \end{array}$$

where

$$\begin{array}{rcl} P' & \triangleq & [x\ is\ 0]\ \overline{c}\langle K_{01}\rangle \mid [x\ is\ 1]\ \overline{c}\langle K_{01}\rangle \\ Q' & \triangleq & [x\ is\ 0]\ \overline{c}\langle K_0\rangle \mid [x\ is\ 1]\ \overline{c}\langle K_1\rangle \end{array}$$

and K, K_{01}, K_0, and K_1 are distinct names. Both processes P and Q generate some secrets, including a key K, send a ciphertext encrypted under

K on a channel c, then receive a message x on the channel c. When x is 0 or 1, the processes reveal a secret, generated at the same time as K and present in the message encrypted under K. The difference between processes P and Q is that process P generates and discloses a single secret K_{01}, whereas process Q generates two different secrets, K_0 and K_1, and may disclose one or the other of them, but not both. The environment cannot detect this difference, since the key K is never revealed. Therefore, we conjecture that P and Q are barbed congruent. On the other hand, framed bisimulation cannot be used to equate P and Q. Intuitively, the definition of framed bisimulation requires that we identify K_{01} with either K_0 or K_1 depending on the value of x, but enables us to choose between K_0 and K_1 only before the value of x is known, with no possibility of backtracking on this choice.

References

[1] M. Abadi. Secrecy by typing in security protocols. In *Theoretical Aspects of Computer Software*, volume 1281 of *Lecture Notes in Computer Science*, pages 611–638. Springer-Verlag, 1997.

[2] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the Fourth ACM Conference on Computer and Communications Security*, pages 36–47, 1997.

[3] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. Technical Report 414, University of Cambridge Computer Laboratory, January 1997. A revised version appeared as Digital Equipment Corporation Systems Research Center report No. 149, January 1998, and an abridged version will appear in Information and Computation.

[4] M. Abadi and A. D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR'97: Concurrency Theory*, volume 1243 of *Lecture Notes in Computer Science*, pages 59–73. Springer-Verlag, 1997.

[5] D. Bolignano. An approach to the formal verification of cryptographic protocols. In *3rd ACM Conference on Computer and Communications Security*, pages 106–118, March 1996.

[6] Data encryption standard. Fed. Inform. Processing Standards Pub. 46, National Bureau of Standards, Washington DC, January 1977.

[7] A. S. Elkjaer, M. Höhle, H. Hüttel, and K. O. Nielsen. Towards automatic bisimilarity checking in the spi calculus. In *Proceedings DMTCS'99+CATS'99*. Springer-Verlag, Singapore, 1999. To appear.

[8] R. Focardi and R. Gorrieri. A classification of security properties. *Journal of Computer Security*, 3(1), 1995.

[9] J. Gray and J. McLean. Using temporal logic to specify and verify cryptographic protocols (progress report). In *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, pages 108–116, 1995.

[10] R. A. Kemmerer. Analyzing encryption protocols using formal verification techniques. *IEEE Journal on Selected Areas in Communications*, 7, 1989.

[11] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer-Verlag, 1996.

[12] C. Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1(1):5–36, 1992.

[13] J. K. Millen. The Interrogator model. In *IEEE Symposium on Security and Privacy*, pages 251–260, 1995.

[14] J. K. Millen, S. C. Clark, and S. B. Freedman. The Interrogator: Protocol security analysis. *IEEE Transactions on Software Engineering*, SE-13(2):274–288, 1987.

[15] R. Milner. *Communication and Concurrency.* Prentice-Hall International, 1989.
[16] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, parts I and II. *Information and Computation*, pages 1–40 and 41–77, September 1992.
[17] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Theoretical Computer Science: 5th GI-Conference, Karlsruhe*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, March 1981.
[18] L. Paulson. Proving properties of security protocols by induction. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 70–83, 1997.
[19] B. Pierce and D. Sangiorgi. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, 6(5):409–453, October 1996.
[20] A. M. Pitts and I. D. B. Stark. Observable properties of higher order functions that dynamically create local names, or: What's new? In *Mathematical Foundations of Computer Science, Proc. 18th Int. Symp., Gdańsk, 1993*, volume 711 of *Lecture Notes in Computer Science*, pages 122–141. Springer-Verlag, 1993.
[21] S. Schneider. Security properties and CSP. In *IEEE Symposium on Security and Privacy*, pages 174–187, 1996.
[22] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc., second edition, 1996.